AN ENHANCED HEART DISEASE PREDICTION TECHNIQUE USING HYBRID MACHINE LEARNING METHODS

A Seminar Report Submitted to

the APJ Abdul Kalam Technological University

in partial Fulfillment of the Requirements for the Award of the Degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

BY

Ms. ATHULYA P D
Ms. ABINA KRISHNA
Ms. ADILA P
Ms. ATHIRA P

Under the Guidance of

Ms. Aparna V K

(Asst. Prof., Dept. of CSE)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VEDAVYASA INSTITUTE OF TECHNOLOGY
KARADPARAMBA, MALAPPURAM NOVEMBER 2019

AN ENHANCED HEART DISEASE PREDICTION TECHNIQUE USING HYBRID MACHINE LEARNING METHODS

A Seminar Report Submitted to

the APJ Abdul Kalam Technological University

in partial Fulfillment of the Requirements for the Award of the Degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

BY

Ms. ATHULYA P D
Ms. ABINA KRISHNA
Ms. ADILA P
Ms. ATHIRA P

Under the Guidance of

Ms. Aparna V K

(Asst. Prof., Dept. of CSE)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VEDAVYASA INSTITUTE OF TECHNOLOGY
KARADPARAMBA, MALAPPURAM
NOVEMBER 2019

DECLARATION

I undersigned hereby declare that the seminar report **“AN ENHANCED HEART DISEASE PREDICTION TECHNIQUE USING HYBRID MACHINE LEARNING METHODS”**, submitted for partial fulfillment of the requirements for the award of degree of Master of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of **Aparna V K**. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: Kozhikode	Signature:
Date:	Name of the Student:

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING VEDAVYASA INSTITUTE OF TECHNOLOGY, KARADPARAMBA

CERTIFICATE

This is to certify that the report entitled **'AN ENHANCED HEART DISEASE PREDICTION TECHNIQUE USING HYBRID MACHINE LEARNING METHODS'** submitted by '**ATHULYA P D**' to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer science and engineering is a bonafide record of the project work carried out by him/her under my/our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Internal Supervisor:

External Supervisor:

Head of the Department:

ACKNOWLEDGEMENT

We thank god, the almighty for the blessing us in making the Project a successful one. The satisfaction that accompanies that the successful completion of any task would be incomplete

without the mention of people whose ceaseless cooperation made it possible, whose constant

guidance and encouragement crown all efforts with success.

We would like to thank our Director **Dr. Arun Korath***, for providing the best facilities and atmosphere for the Project completion and presentation.*

We express our heartfelt gratitude towards **Dr. Sangheethaa.S***, Principal, Vedavyasa Institute of Technology, for extending all the facilities required for doing this work.*

We express our sincere gratitude to **Prof. Mrs. S. Kavitha Murugesan***, Head of Department, Dept. of Computer Science &amp; Engineering, Vedavyasa Institute of Technology, Malappuram for her guidance and support.*

We are grateful to our project guide **Mrs. Aparna V K** *& Class tutor* **Mrs. Sruthi M P***, Asst. Professor, Dept. of Computer Science & Engineering. For the guidance, inspiration and constructive suggestions that helpful us in the preparation of this work and all lecturer of department of computer science and engineering for their valuable suggestions.*

Last but not the least; we extend heartfelt gratitude to our parents and friends for their support and assistance.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CSV	COMMA-SEPARATED VALUES
DFD	DATA FLOW DIAGRAM
HRFLM	HYBRID RANDOM FOREST LINEAR MODEL
UCI	UNIVERSITY OF CALIFORNIA, IRVINE
SQL	STRUCTURED QUERY LANGUAGE

ABSTRACT

Heart disease is one of the significant causes of death today. Prediction of heart disease is a critical challenge. Machine Learning helps to make decisions and predictions from the large quantity of data. These data are produced by the healthcare industry. It is used in different areas of Internet of Things. Thus we use different techniques of Machine Learning to predict heart disease. A more accurate prediction of cardiovascular disease can be made by applying Hybrid Random Forest with Linear Model.

Keywords - *Machine learning, heart disease prediction, feature selection, prediction model, classification algorithms, cardiovascular disease (CVD)*

CHAPTER 1

INTRODUCTION

The aim of this project is to predict heart disease using machine learning algorithms. This project combines characteristics of Random Forest (RF) and Linear Method (LM) to investigate their performance on heart disease datasets obtained from the UCI data repository. We produce an enhanced performance level with an accuracy level of 88.7% through the prediction model for heart disease with the hybrid random forest with a linear model (HRFLM). It supports following features:

- Login/ Sign Up
- Viewing patient's history of prediction
- User can enter the values of various parameters on the basis of which his risk factor will be calculated using machine learning algorithms.

1.1 BACKGROUND

Among all fatal disease, heart attacks diseases are considered as the most prevalent. Medical practitioners conduct different surveys on heart diseases and gather information of heart patients, their symptoms and disease progression. Increasingly are reported about patients with common diseases who have typical symptoms. In this fast moving world people want to live a very luxurious life so they work like a machine in order to earn lot of money and live a comfortable life therefore in this race they forget to take care of themselves, because of this there food habits change their entire lifestyle change, in this type of lifestyle they are more tensed they have blood pressure, sugar at a very young age and they don't give enough rest for themselves and eat what they get and they even don't bother about the quality of the food if sick the go for their own medication as a result of all these small negligence it leads to a major threat that is the heart disease The term 'heart disease' includes the diverse diseases that affect heart. The number of people suffering from heart disease is on the rise. The report from world health organization shows us a large number of people that die every year due to the heart disease all over the world. Heart disease is also stated as one of the greatest killers in Africa.

Machine learning in medicine has recently made headlines. Google has developed a machine learning algorithm to help identify cancerous tumors on mammograms. Stanford is using a deep learning algorithm to identify skin cancer. A recent JAMA article reported the results of a deep machine-learning algorithm that was able to diagnose diabetic retinopathy in retinal images. It’s clear that machine learning puts another arrow in the quiver of clinical decision making.

Machine Learning proves to be crucial in the medical field. It provides a meaningful base to critical decisions. It helps to create a complete study proposal. One of the most important uses of machine learning is that it helps in keeping human bias away from medical conclusion with the help of proper statistical treatment. By use of machine learning for exploratory analysis because of nontrivial information in large volumes of data. The health care industries collect huge amounts of data that contain some hidden information, which is useful for making effective decisions for providing appropriate results and making 2 effective decisions on data, some machine techniques are used to better the experience and conclusion that have been given. This project will use the machine learning knowledge to give a user-oriented approach to new and hidden patterns in the data. The knowledge which is implemented can be used by the healthcare experts to get better quality of service and to reduce the extent of adverse medicine effect.

1.2 PROBLEM STATEMENT

Heart disease can be managed effectively with a combination of lifestyle changes, medicine and, in some cases, surgery. With the right treatment, the symptoms of heart disease can be reduced and the functioning of the heart improved. The predicted results can be used to prevent and thus reduce cost for surgical treatment and other expensive.

The overall objective of this project will be to predict accurately with few tests and attributes the presence of heart disease. Attributes considered form the primary basis for tests and give accurate results more or less. Many more input attributes can be taken but our goal is to predict with few attributes and faster efficiency the risk of having heart disease. Decisions are often made based on doctors' intuition and experience rather than on the knowledge rich data hidden in the data set and databases. This practice leads to unwanted biases, errors and excessive medical costs which affects the quality of service provided to patients.

Machine learning holds great potential for the healthcare industry to enable health systems to systematically use data and analytics to identify inefficiencies and best practices that improve care and reduce costs. The opportunities to improve care and reduce costs concurrently could apply to as much as 30% of overall healthcare spending. The successful application of machine learning in highly visible fields like e-business, marketing and retail has led to its application in other industries and sectors. Among these sectors just discovering is healthcare. The healthcare environment is still "information rich" but "knowledge poor‟. There is a wealth of data available within the healthcare systems.

1.3 OBJECTIVE

The main objective of this research is to develop a heart disease prediction system. The system can discover and extract hidden knowledge associated with diseases from a historical heart data set. Heart disease prediction system aims to exploit machine learning techniques on medical data set to assist in the prediction of the heart diseases.

- Provides new approach to concealed patterns in the data.
- Helps avoid human biasness.
- To implement HRFLM that classifies the disease as per the input of the user.
- Reduce the cost of medical tests.

Clinical decisions are often made based on doctor’s insight and experience rather than on the knowledge rich data hidden in the dataset. This practice leads to unwanted biases, errors and excessive medical costs which affects the quality of service provided to patients. The proposed system will integrate clinical decision support with computer-based patient records (Data Sets). This will reduce medical errors, enhance patient safety, decrease unwanted practice variation, and improve patient outcome. This suggestion is promising as data modeling and analysis tools, e.g., machine learning, have the potential to generate a knowledge rich environment which can help to significantly improve the quality of clinical decisions. There are voluminous records in medical data domain and because of this, it has become necessary to use machine learning techniques to help in decision support and prediction in the field of healthcare. Therefore, medical machine learning contributes to business intelligence which is useful for diagnosing of disease

1.4 SCOPE

Here the scope of the project is that integration of clinical decision support with computer-based patient records could reduce medical errors, enhance patient safety, decrease unwanted practice variation, and improve patient outcome. This suggestion is promising as tools, such as machine learning, have the potential to generate predictions which can help to significantly improve the quality of clinical decisions.

1.5 LIMITATIONS

Medical diagnosis is considered as a significant yet intricate task that needs to be carried out precisely and efficiently. The automation of the same would be highly beneficial. Clinical decisions are often made based on doctor’s intuition and experience rather than on the knowledge rich data hidden in the database. This practice leads to unwanted biases, errors and excessive medical costs which affects the quality of service provided to patients. Machine learning has the potential to generate predictions which can help to significantly improve the quality of clinical decisions.

1.6 SUMMARY

Introduction has given an overview about heart disease prediction and its importance. The Heart Disease Prediction App will let user enter the values of various parameters on the basis of which his risk factor will be calculated using machine learning algorithms. Further chapters discuss the related work, existing system, proposed system and conclusion.

CHAPTER 2

RELATED WORKS

2.1 EFFECTIVE DIAGNOSIS OF HEART DISEASE THROUGH NEURAL NETWORKS ENSEMBLES

- R. Das, I. Turkoglu, A. Sengur

Heart disease, which is usually called coronary artery disease (CAD), is a broad term that can refer to any condition that affects the heart. CAD is a chronic disease in which the coronary arteries gradually harden and narrow and the most common form of cardiovascular disease in the United States and the leading cause of heart attacks. Moreover, cardiovascular disease is the leading killer of American women, causing almost 500,000 deaths every year. While many people with heart disease have symptoms such as chest pain and fatigue, as many as 50% have no symptoms until a heart attack occurs. According to the American heart association (AHA), CAD is the leading killer of American men and women, responsible for more than one of every five deaths in 2001. Many statistics show CAD as the leading cause of premature and permanent disability among American workers. There are many risk factors associated with CAD. Some risk factors for CAD, such as your sex, age and family history, cannot be changed. Other risk factors for CAD that are related to lifestyle often can be changed. For example, smoking, high cholesterol, high blood pressure and physical inactivity are all risk factors for coronary artery disease that can be modified and in some cases eliminated with lifestyle changes and medication. Diabetes and obesity can sometimes be prevented when lifestyle changes are made early in life. Having so many factors to analyze to diagnose the heart diseases, physicians generally make decisions by evaluating the current test results of the patients. The previous decisions made on other patients with the same condition are also examined by the physicians. These complex procedures are not easy when considering the number of factors that the physician has to evaluate. So, diagnosing the heart disease of a patient involves experience and highly skilled physicians. Recent advances in the field of artificial intelligence have led to the emergence of expert systems for medical applications. Moreover, in the last few decades computational tools have been designed to improve the experiences and abilities of physicians for making decisions about their patients. Motivated by the need of such an expert system, in this study, we propose a method to efficiently diagnose the heart disease. The proposed system uses neural networks ensemble model. Ensemble based methods enable an increase in generalization performance by combining several individual neural networks train on the same task. We realized our proposal with SAS base software 9.1.3 (license number: 291468). SAS enterprise miner streamlines the entire data mining process from data access to model assessment. It supports all necessary tasks within a single, integrated solution while providing the flexibility for efficient collaborations. This paper obtained 89.01% classification accuracy from the experiments made on the data taken from Cleveland heart disease database. They also obtained 80.95% and 95.91% sensitivity and specificity values, respectively, in heart disease diagnosis. The remaining of the paper is organized as follows. This paper presents a brief overview on related work, dataset description, neural networks and ensemble based methods in the background section.

2.2 HEART FUNCTION MONITORING, PREDICTION AND PREVENTION OF HEART ATTACKS: USING ARTIFICIAL NEURAL NETWORKS

- D. K. Ravish, K. J. Shanthi, N. R. Shenoy

Heart is the most important component of the cardiovascular system. It weighs around 250-350 grams i.e. about the size of a fist. It beats around 2.5 billion times during as lifespan of 66-68 years. The heart is electrically stimulated by a special unit called Sino atrial node. This region produces a definite potential and slowly discharges, thus sending an electrical impulse across the atria. This electrical impulse is very sequential in nature and responsible for the systole and diastole in the four chambers respectively. Disturbance in the electrical activity of the heart leads to a condition called Arrhythmia. Arrhythmia is a diseased condition of the heart in which there is a great variation of the heart's rhythm from the normal sinus rhythm. It may be fast (i.e. tachycardia) or slow (i.e. bradycardia) in nature. Some of common types of arrhythmias are a) Tachycardia, b) Bradycardia, c) Skipped beat or pause, d) Atrial, ventricular fibrillation, e) Pre mature ventricular contraction (PVC) and f) Rhythm and electrical conduction disorders etc. For precise and accurate treatment of arrhythmias we require combination of two data's i.e. ECG data and clinical data. The procedure involved in the data acquisition, processing, analysis and transmission of these data's is very expensive. This cost factor severely affects the quality and standards of medical care. Annual health reports by W.H.O shows that around 17.3 million people died from heart attacks in 2008 worldwide. Cardio vascular heart diseases are also leading cause of death in India, accounting for an estimated 2.25 million deaths in 2010 alone. This shows that Heart attacks are the number one cause of death globally and more people die from heart attacks compared to any other disease. Statistical and adaptive learning models are the two main approaches that have been applied to predict heart diseases based on ECG and clinical databases. Data mining and statistical analysis is a very powerful tool for distinguishing a diseased person from a healthy person. They utilize the hidden medical information in the vast patient databases and classify the heart disease into various types and categories. Thus this forms the foundation for any therapy or treatment to be given. This paper necessitates an efficient system or a model that can predict heart attacks as the inputs are highly uncertain, non linear and continuously varying irrespective of time. Also each input is unique and different from the other inputs. This accounts for the need of highly non linear system that is flexible and adaptive in nature. Thus we use Artificial Neural Network (ANN) for the prediction and declaration of the results based on the inputs fed to it. Some of the features of artificial neural networks are: a) Non linearity b) Input- output mapping c) Adaptivity d) Fault tolerance e) Uniformity of analysis and design f) Neurobiological analogy g) Contextual information. Some of the other features are it is highly precise, very accurate and has a good learning rate. Due to these superior features we have chosen artificial neural network as the best tool for predicting and diagnosing the heart diseases. In this paper we have described five crucial steps taken in predicting heart diseases. They are Data acquisition stage, Processing stage, Network training stage, Disease predicting stage and finally Data transmission stage.

2.3 PREDICTION OF HEART DISEASE USING BACK PROPAGATION MLP ALGORITHM

- M. Durairaj, V. Revathi

Heart is an important organ of all living creature, which plays a vital role of pumping blood to the rest of the organs through the blood vessels of the circulatory system. Any functional problem in the heart has a direct impact on the survival of concerned human being, since it affects other parts of the body such as brain, lungs, kidney, liver etc. Heart Diseases describe a range of conditions that affect the heart and stand as a leading cause of death all over the world. The clinical symptoms of the Heart Disease complicate the prognosis, as it is influenced by many factors like functional and pathologic appearance. This could subsequently delay the prognosis of the disease. Hence, there is a need for the invention of newer concepts to improve the prediction accuracy with short span. Disease prognosis through numerous factors or symptoms is a multilayered problem, even that could lead to a false assumption. Therefore, an attempt is made to bridge the knowledge and the experience of the experts and to build a system that fairly supports the diagnosing process. Hence, this paper proposes a new approach, by implementing the Multilayer Perceptron (MLP) Algorithm over a Heart Disease dataset that was taken from the popular UCI machine learning repository.

2.4 AN INTEGRATED DECISION SUPPORT SYSTEM BASED ON ANN AND FUZZY AHP FOR HEART FAILURE RISK PREDICTION

- O. W. Samuel, G. M. Asogbon

One of the most common and complex health problems around the world to date is heart failure (HF) which usually occurs when the heart fails to pump sufficient amount of blood to meet the needs of the body. The HF patients often experience some symptoms such as shortness of breath, weakness of the body, and swollen feet. The findings from the previous investigations have identified HF and its resulting complications as one of the major causes of reduced quality of life as well as mortality in many technologically advanced countries. The management of HF disease is quite complex and even worse in developing countries where there are rare possibility of adequate diagnostic tools and limited number of medical experts especially in some rural communities. Therefore, accurately predicting the HF risk in patients/individuals would be essential for reducing their associated risks of severe heart problems and increasing their safety and productivity during daily life activities. The conventional methods for the diagnosis of HF risks are mainly based on the analysis of patients' medical history, physical examination report, and review of relevant symptoms by a medical practitioner, which often lead to imprecise diagnosis and sometimes delay in diagnosis outcome because of human errors. In an attempt to resolve these issues, medical decision support systems based on emerging computational methods such as support vector machine, k-nearest neighbour, decision tree, fuzzy logic, and artificial neural network (ANN) have been proposed in previous studies for HF-risks prediction.

2.5 MACHINE LEARNING TECHNIQUES FOR HEART DISEASE PREDICTION

- A. Lakshmanarao,Y.Swathi, P.Sri Sai Sundareswar

The most common type of heart disease is Coronary heart disease (CHD). More than 3,50,000 people are dying with this disease annually. In the United States every year 610000 people are dying with heart disease. Every year 7,35,000 members having a heart attack, out of which 5,25,000 members are 1st heart attack and 2,10,000 members who have already had a heart attack [5]. In Asian countries also, 22% of the deaths (in total heart disease deaths) are due to heart diseases. The contributing factors for heart disease are more (blood pressure, diabetes, current smoker, high cholesterol, etc..). So it is not easy to identify heart disease. Various techniques in data mining and neural networks have been employed to find out the severity of heart disease among humans. The nature of CHD illness is perplexing, what's more, subsequently; the malady must be taken care of cautiously. Not doing early detection, may influence the heart or cause unexpected passing. The point of view of medicinal science and information digging are utilized for finding different sorts of metabolic disorders. Machine Learning is a technique that helps the system to learn from previous data samples, examples without being explicitly programmed. Machine learning creates logic based on historical data. Machine Learning plays a vital role in many fields. It also shows its impact on heart disease detection. Deep Learning is a part AI, which can also be considered as a subset of machine learning. Deep learning can also be applied to the number of research areas. It is also applied for heart disease prediction.

2.6 HEART DISEASE PREDICTION SYSTEM USING DATA MINING TECHNIQUES

- V. Krishnaiah, G. Narsimha

Heart disease is that the biggest reason behind death nowadays. Blood pressure, cholesterol, pulse rate are the major reason for the heart disease. The heart is an operating system of our human body. If the function of heart is not done properly it affect other body part also. There are some risks factors of heart disease such as Family history, High blood pressure, Cholesterol, Age, Poor diet, Smoking. When blood vessels are overstretched, the risk levels of the blood vessels are increased. This leads to the blood pressure. Blood pres- sure is typically measured in terms of systolic and diastolic. Systolic shows the pressure in the arteries when the heart muscle contracts. Diastolic indicates the pressure in the arteries when the heart muscle is in resting state. The level of fats increased in the blood are causes the heart disease. The lipids are in the arteries hence the arteries become narrow and blood flow is also become slow. Age is the non-modifiable risk factor which also a reason for heart disease. Smoking is the reason 35% of the death of heart diseases. Because it limits the oxygen level in the blood then it damage and tighten the blood vessels. Different data mining techniques such as Nave Bayes, KNN algorithm, Decision tree, Neural Network are used to predict the risk of heart disease. The Nave Bayes method is used to predict the heart disease through probability. Improved k means is used to provide the classified report for the heart disease. In many hospitals, more experience doctors are there. They know how analyze the patient within a given period. But in some hospitals new doctors don’t have more experience. They can’t take quickly decision for e.g. heart disease prediction so they use this system for easy and quicker decision. In medical college, students who study about how to diagnose heart i.e. PHD students. So it easy to learn using this heart disease system for those students.

2.7 STUDY OF HEART DISEASE DIAGNOSIS BY COMPARING VARIOUS CLASSIFICATION ALGORITHMS

- C. Sowmiya and P. Sumitra

According to World Health Organization, Hypertensive heart attack is one of the prime causes of death. Health Care industry has huge amount of data, which is not mined. It has hidden patterns which is necessary for prediction of heart disease risk. We know, heart is very important part of human body. If, the organs of the body that is brain and heart there is an insufficiency in blood circulation, heart stop working immediately and it causes death. The risk parameters associated with heart risk are age, family history, hypertension, high cholesterol, diabetes, smoking, tobacco, alcohol consumption, obesity, poor diet and chest pain. Even about sixty percent of total population are suffering from the heart disease, so detection of the heart disease earlier can prevent the heart disease. In many cases heart disease almost noticed at the final stage or after death. It's difficult to cure heart disease at the final stages so people are very reluctant to treat at the early stages of heart disease. Health care industries have very huge amount of data and which is not mined to discover hidden pattern. Solution to this problem is data mining techniques. It is the process of analysis of large dataset and extraction useful information from it. Mining techniques are:

1. Association Mining
2. Classification Techniques
3. Clustering Techniques

Association rule mining is a method to discover interesting relationship between variables in large set of data. Classification is used to extract a model describing useful classes. There are many classification techniques which are used in diagnosis of heart disease, like Decision tree, Naïve Bayes, SVM, K- Nearest neighbour and many more. Clustering is procedure of grouping similar features datasets into cluster. K-mean algorithm is used for clustering. It is faster than other. In this paper different kind of classification methods which are applied in the forecasting of heart disease has been talked and also comparison made on the classifiers to justify which algorithm achieve the high accuracy.

CHAPTER 3

METHODOLOGY

Medical diagnosis is considered as a significant yet intricate task that needs to be carried out precisely and efficiently. The automation of the same would be highly beneficial. Clinical decisions are often made based on doctor’s intuition and experience rather than on the knowledge rich data hidden in the database. This practice leads to unwanted biases, errors and excessive medical costs which affects the quality of service provided to patients.
In this project, we develop an enhanced Heart Disease Prediction System combining the characteristics of Random Forest and Linear Method.

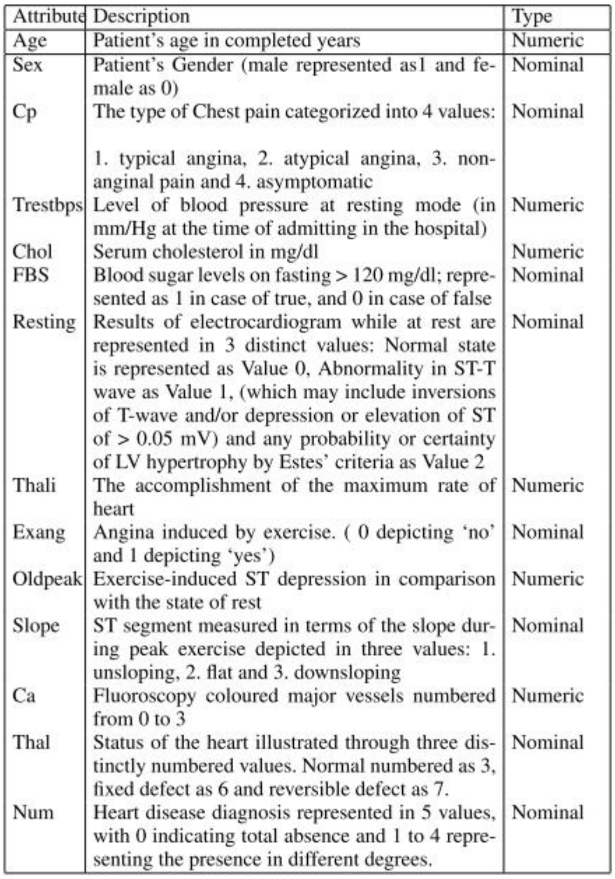
The proposed system performs heart disease classification on the Cleveland UCI repository. ML process starts from a pre-processing data phase followed by feature selection based on DT entropy, classification of modelling performance evaluation, and the results with improved accuracy. The feature selection and modelling keep on repeating for various combinations of attributes. The performance of each model generated based on 13 features and ML techniques used for each iteration and performance are recorded.

3.1. DATA PRE-PROCESSING

Heart disease data is pre-processed after collection of various records. The dataset contains a total of 303 patient records, where 6 records are with some missing values. Those 6 records have been removed from the dataset and the remaining 297 patient records are used in pre-processing. The multi-class variable and binary classification are introduced for the attributes of the given dataset. The multi-class variable is used to check the presence or absence of heart disease. In the instance of the patient having heart disease, the value is set to 1, else the value is set to 0 indicating the absence of heart disease in the patient. The pre-processing of data is carried out by converting medical records into diagnosis values. The results of data pre-processing for 297 patient records indicate that 137 records show the value of 1 establishing the presence of heart disease while the remaining 160 reflected the value of 0 indicating the absence of heart disease.

3.2. FEATURE SELECTION AND REDUCTION

Table 3.1 shows the UCI dataset detailed information with attributes used.

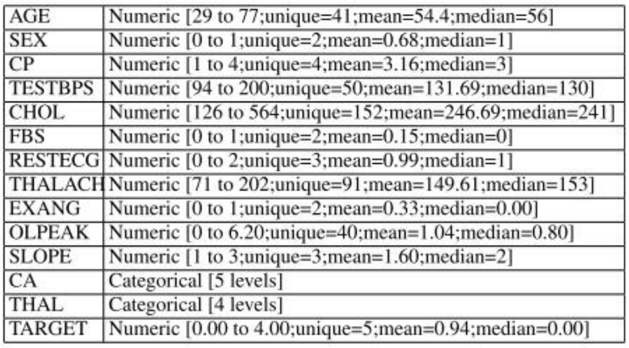


Attribute	Description	Type
Age	Patient's age in completed years	Numeric
Sex	Patient's Gender (male represented as1 and female as 0)	Nominal
Cp	The type of Chest pain categorized into 4 values: 1. typical angina, 2. atypical angina, 3. non-anginal pain and 4. asymptomatic	Nominal
Trestbps	Level of blood pressure at resting mode (in mm/Hg at the time of admitting in the hospital)	Numeric
Chol	Serum cholesterol in mg/dl	Numeric
FBS	Blood sugar levels on fasting > 120 mg/dl; represented as 1 in case of true, and 0 in case of false	Nominal
Resting	Results of electrocardiogram while at rest are represented in 3 distinct values: Normal state is represented as Value 0, Abnormality in ST-T wave as Value 1, (which may include inversions of T-wave and/or depression or elevation of ST of > 0.05 mV) and any probability or certainty of LV hypertrophy by Estes' criteria as Value 2	Nominal
Thali	The accomplishment of the maximum rate of heart	Numeric
Exang	Angina induced by exercise. (0 depicting 'no' and 1 depicting 'yes')	Nominal
Oldpeak	Exercise-induced ST depression in comparison with the state of rest	Numeric
Slope	ST segment measured in terms of the slope during peak exercise depicted in three values: 1. unsloping, 2. flat and 3. downsloping	Nominal
Ca	Fluoroscopy coloured major vessels numbered from 0 to 3	Numeric
Thal	Status of the heart illustrated through three distinctly numbered values. Normal numbered as 3, fixed defect as 6 and reversible defect as 7.	Nominal
Num	Heart disease diagnosis represented in 5 values, with 0 indicating total absence and 1 to 4 representing the presence in different degrees.	Nominal

Table 3.1 UCI dataset attributes detailed information

From among the 13 attributes of the data set, two attributes pertaining to age and sex are used to identify the personal information of the patient. The remaining 11 attributes are considered important as they contain vital clinical records. Clinical records are vital to diagnosis and learning the severity of heart disease. As previously mentioned in this experiment, several

(ML) techniques are used namely, NB, GLM, LR, DL, DT, RF, GBT and SVM. The experiment was repeated with all the ML techniques using all 13 attributes.
Table 3.2 shows the data type and range of values.



AGE	Numeric [29 to 77;unique=41;mean=54.4;median=56]
SEX	Numeric [0 to 1;unique=2;mean=0.68;median=1]
CP	Numeric [1 to 4;unique=4;mean=3.16;median=3]
TESTBPS	Numeric [94 to 200;unique=50;mean=131.69;median=130]
CHOL	Numeric [126 to 564;unique=152;mean=246.69;median=241]
FBS	Numeric [0 to 1;unique=2;mean=0.15;median=0]
RESTECG	Numeric [0 to 2;unique=3;mean=0.99;median=1]
THALACH	Numeric [71 to 202;unique=91;mean=149.61;median=153]
EXANG	Numeric [0 to 1;unique=2;mean=0.33;median=0.00]
OLPEAK	Numeric [0 to 6.20;unique=40;mean=1.04;median=0.80]
SLOPE	Numeric [1 to 3;unique=3;mean=1.60;median=2]
CA	Categorical [5 levels]
THAL	Categorical [4 levels]
TARGET	Numeric [0.00 to 4.00;unique=5;mean=0.94;median=0.00]

Table 3.2 UCI dataset range and datatype.

3.3. CLASSIFICATION MODELLING

The clustering of datasets is done on the basis of the variables and criteria of Decision Tree (DT) features. Then, the classifiers are applied to each clustered dataset in order to estimate its performance. The best performing models are identified from the above results based on their low rate of error. The performance is further optimized by choosing the DT cluster with a high rate of error and extraction of its corresponding classifier features. The performance of the classifier is evaluated for error optimization on this data set.

3.3.1. Decision Trees

For training samples of data D, the trees are constructed based on high entropy inputs. These trees are simple and fast constructed in a top down recursive divide and conquer (DAC) approach. Tree pruning is performed to remove the irrelevant samples on D.

$$Entropy = -\sum_{j=1}^{m} p_{ij} \log_2 p_{ij}$$

3.3.2. Language Model

For given input features x_i , y_i with input vector x_i of data D the linear form of solution f(x)=mx+b is solved by subsequent parameters:

$$m = \frac{\left(\sum_i x_i y_i\right) - n\overline{x_i}\overline{y_i}}{\left(\sum_i x_i^2\right) - n\overline{x_i}^2}$$

b = ¯y − m¯x where ¯x , ¯y are the means.

3.3.3. Support Vector Machine

Let the training samples having dataset Data ={y_i,x_i}; i =1 , 2 , . . . , n where $x_i \in R_n$ represent the i^{th} vector and $y_i \in R_n$ represent the target item. The linear SVM finds the optimal hyperplane of the form f (x) = w^Tx + b where w is adimensional coefficient vector and b is a offset. This is done by solving the subsequent optimization problem:

$$Min_{w,b,\xi_i} \frac{1}{2}w^2 + C\sum_{i=1}^{..}\xi_i$$
$$s.t.\ y_i\left(w^T x_i + b\right) \geq 1 - \xi_i, \xi_i \geq 0, \quad \forall_i \in \{1, 2, \ldots, m\}$$

3.3.4. Random Forest

This ensemble classifier builds several decision trees and incorporates them to get the best result. For tree learning, it mainly applies bootstrap aggregating or bagging. For a given data, X = {x_1 , x_2 , x_3 , ..., x_n} with responses Y ={x_1 , x_2 , x_3 , ..., x_n}which repeats the bagging from b = 1 to B . The unseen samples x’ is made by averaging the predictions from every individual trees on x’:

$$j = \frac{1}{B}\sum_{b=1}^{B} fb\left(x'\right)$$

The uncertainty of prediction on these tree is made through its standard deviation,

$$\sigma = \sqrt{\frac{\sum_{\mathrm{b=1}}^{\mathrm{B}} \left(fb\left(x'\right) - \hat{f}\right)^2}{B-1}}$$

3.3.5. Naïve Bayes

This learning model applies Bayes rules through independent features. Every instance of data D is allotted to the class of highest subsequent probability. The model is trained through the Gaussian function with prior probability $P(X_f)$=priority ∈ (0 : 1)

$$\begin{aligned} &P\left(X_{f1}, X_{f2}, \ldots, X_{f_n}|c\right) \\ &\quad = \prod_{i=1}^{n} P\left(X_{fi}|c\right) \\ &P\left(X_f|c_i\right) \\ &\quad = \frac{P\left(c_i|X_f\right) P\left(X_f\right)}{P\left(c_i\right)} \quad c \in \{\, benign,\ malignant \,\} \end{aligned}$$

At last, the testing data is categorized based on the probability of association:

$$c_{nb} = \arg\max P\left(c_k\right) \prod_{i=1}^{n} P\left(X_{fi}|c_k\right), \quad for\ k = 1, 2$$

3.3.6. Neural Networks

The neuron components includes inputs x_i , hidden layers and output y_i. The final result is produced through the activation function like sigmoid and a bias constant b.

$$f\left(b + \sum_{i=1}^{n} x_i u_i\right)$$

3.3.7. K-Nearest Neighbour

It extract the knowledge based on the samples Euclidean distance function $d(x_i , x_j)$and the majority of k-nearest neighbors.

$$d\left(x_{i,x_{i,}}\right) = \sqrt{\left(x_{i,1} - x_{j,1}\right)^2 + \cdots + \left(x_{i,m} - x_{j,m}\right)^2}$$

3.4. PERFORMANCE MEASURES

Several standard performance metrics such as accuracy, pre- cision and error in classification have been considered for the computation of performance efficacy of this model. Accuracy in the current context would mean the percentage of instances correctly predicting from among all the available instances. Precision is defined as the percentage of corrective prediction in the positive class of the instances. Classification error is defined as the percentage of accuracy missing or error available in the instances. To identify the significant features of heart disease, three performance metrics are used which will help in better understanding the behavior of the various ML technique focuses on the best performing model compared to the existing models. We introduce HRFLM, which produces high accuracy and less classification error in the prediction of heart disease. The performance of every classifier is evaluated individually and all results are adequately recorded for further investigation.

CHAPTER 4

SYSTEM ARCHITECTURE

In this system we are implementing effective heart attack prediction system using HRFLM algorithms. We can give the input as in CSV file or manual entry to the system. After taking input the algorithms apply on that input that is HRFLM. After accessing data set the operation is performed and effective heart attack level is produced. The proposed system will add some more parameters significant to heart attack with their weight, age and the priority levels are by consulting expertise doctors and the medical experts. The heart attack prediction system designed to help the identify different risk levels of heart attack like normal, low or high and also giving the prescription details with related to the predicted result.

Figure 4.1 shows the proposed system architecture.

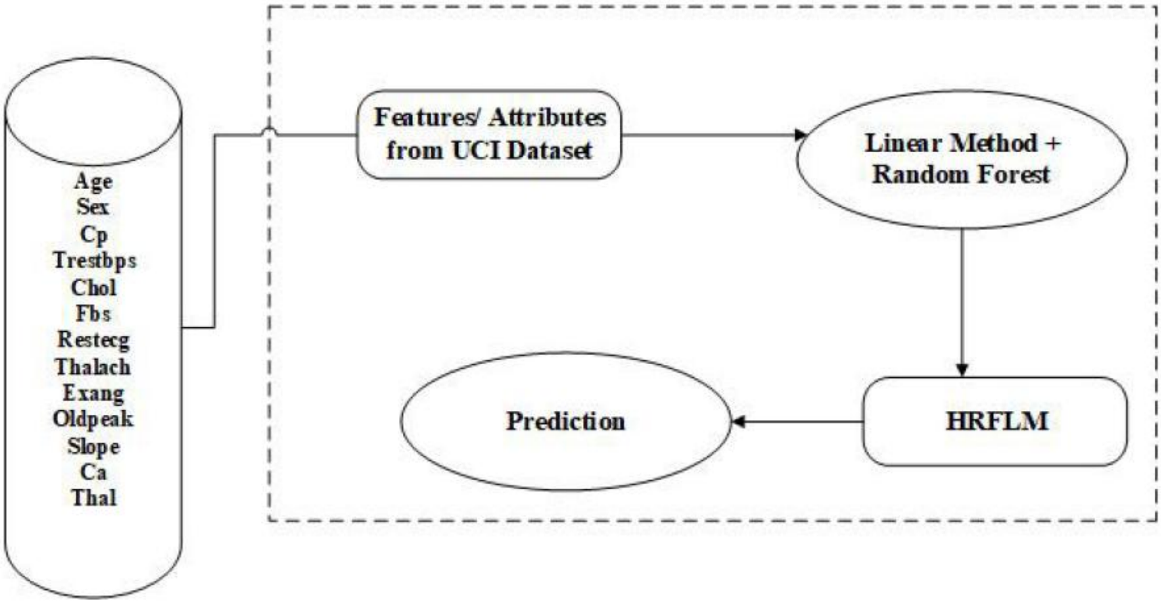


Fig 4.1 Proposed System Architecture

4.1. REQUIREMENT ANALYSIS

Requirements analysis understands the users requirements within the framework of the organization objective and the environment in which the system being installed. Requirements analysis is a software engineering task that bridges the gap between system level software allocation and software design. The first step in designing a system was to identify the underlying cause behind the problems in the existing system. Requirement study has been done to the gather the required information. Requirements analysis enables the system engineer to specify software function and performance, indicate software's interface with other system elements, and establish constraints that software must meet. Requirements analysis allows the software engineer to refine the software allocation and build models of the data, Requirements analysis provides the software designer with models that can be translated into data, architectural, interface and procedural design. Finally, the requirement specification provides the developer and the customer with the means to assess quality once software is built and behavioural domains that will be treated by software.

4.2. SOFTWARE LIFECYCLE

Requirements analysis enables the system engineer to specify software function and performance, indicate software's interface with other system elements, and establish constraints that software must meet. Requirements analysis allows the software engineer to refine the software allocation and build models of the data, functional and behavioural domains that will be treated by software. Requirements analysis provides the software designer with models that can be translated into data, architectural, interface, and procedural design. Finally, the requirement specification provides the developer and the customer with the means to assess quality once software is built.

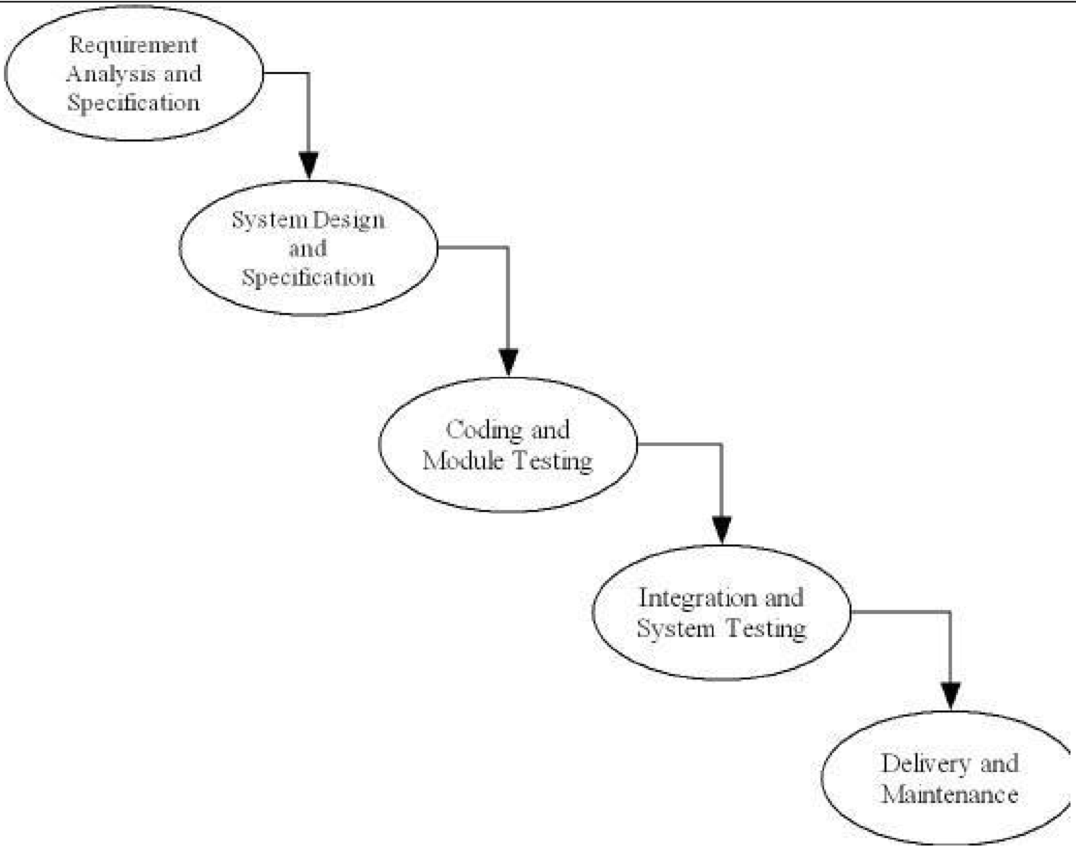

Requirement Analysis and Specification
System Design and Specification
Coding and Module Testing
Integration and System Testing
Delivery and Maintenance

Fig 4.7 Software Lifecycle

It is a four-step process, which includes -

1. Feasibility Study
2. Requirement Elicitation and Analysis
3. Software Requirement Specification
4. Software Requirement Validation

4.3. FEASIBILTY STUDY

Feasibility is defined as the practical extent to which a project can be performed successfully. To evaluate feasibility, a feasibility study is performed, which determines whether the solution considered to accomplish the requirements is practical and workable in the software. Information such as resource availability, cost estimation for software development, benefits of the software to the organization after it is developed and cost to be incurred on its maintenance are considered during the feasibility study. The objective of the feasibility study is to establish the reasons for developing the software that is acceptable to users, adaptable to change and conformable to established standards. Various other objectives of feasibility study are listed below:

- To analyze whether the software will meet organizational requirements.
- To determine whether the software can be implemented using the current technology and within the specified budget and schedule.
- To determine whether the software can be integrated with other existing software.

4.3.1. Technical Feasibility

Technical feasibility assesses the current resources (such as hardware and software) and technology, which are required to accomplish user requirements in the software within the allocated time and budget. For this, the software development team ascertains whether the current resources and technology can be upgraded or added in the software to accomplish specified user requirements. Technical feasibility also performs the following tasks:

- Analyzes the technical skills and capabilities of the software development team members.
- Determines whether the relevant technology is stable and established.
- Ascertains that the technology chosen for software development has a large number of users so that they can be consulted when problems arise or improvements are required.

4.3.2. Operational Feasibility

Operational feasibility assesses the extent to which the required software performs a series of steps to solve business problems and user requirements. This feasibility is dependent on human resources (software development team) and involves visualizing whether the software will operate after it is developed and be operative once it is installed. Operational feasibility also performs the following tasks:

- Determines whether the problems anticipated in user requirements are of high priority.
- Determines whether the solution suggested by the software development team is acceptable.
- Analyzes whether users will adapt to a new software.
- Determines whether the organization is satisfied by the alternative solutions proposed by the software development team.

4.3.3. Economic Feasibility

Economic feasibility determines whether the required software is capable of generating financial gains for an organization. It involves the cost incurred on the software development team, estimated cost of hardware and software, cost of performing feasibility study, and so on. For this, it is essential to consider expenses made on purchases (such as hardware purchase) and activities required to carry out software development. In addition, it is necessary to consider the benefits that can be achieved by developing the software. Software is said to be economically feasible if it focuses on the issues listed below:

- Cost incurred on software development to produce long-term gains for an organization.
- Cost required to conduct full software investigation (such as requirements elicitation and requirements analysis).
- Cost of hardware, software, development team, and training.

4.3.4 Behavioural Feasibility

Normal human psychology of human beings indicate that people are resistant to change and computers are known to facilitate change. Any project formulations should consider this factor also. Before the development of the Project titled "Delhi Metro", the need to study the feasibility of the successful execution of the project was felt and thus the following factors are considered for a Feasibility Study:

1. Need Analysis.
2. Provide the users information pertaining to the preceding requirement.

4.4. FUNCTIONAL REQUIREMENTS

The statement of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situation is analysed here. The functional requirements are:

4.4.1. User Android Application

- User will have to login to application with email id and Password.
- There will be option for entering patient details.
- Users can then enter details vital to diagnosis.
- Users can also view prediction history.

4.4.2. Website

- Staff can login and view all details of patients.
- The staff can delete patient records.
- They can also view prediction history of all patients.

4.6. NON-FUNCTIONAL REQUIREMENTS

Non-functional requirement is a requirement that specifies criteria that can be used ti judge the operation of a system, rather than specific behaviours. Non-functional requirements are often called qualities of a system. The Non-functional requirements includes the digital data of the organization. The memory requirement is minimum 2GB RAM and 20GB of secondary storage.

4.7. COST/BENEFIT ANALYSIS

Cost-benefit analysis compares the expected financial gain derived from a particular set of actions with the expected cost of providing each action to determine the most profitable option. The projected benefits of a plan or program are divided by its estimated total long-term cost. Additionally, subjective variables such as customer satisfaction are often quantified and factored in. The program must take into account the fact that the impacts of costs and the values of benefits might extend far into the future.

4.8. PROCESS ANALYSIS

Process analysis is a detailed study of process that looks at everything that is going on in the process. Process Analysis is the documentation and detailed understanding of how work is performed Goals of process Analysis are:

- Reduce the time, cost and effort of the process.
- Improve process efficiency.

4.8.1. Problem Definition

The purpose of the project entitled "**Heart Disease predictiom**" is to give predictions as per the 14 attributes given by the user. As a result, Doctors can use this application to guide their decision regarding heart disease.

4.8.2. Module Description

The main modules in the task include:

1. User

2. View History

3. Database

1. User

- User need to login to the app by Email id and provided Password.
- In the app, user can manually enter details to register.
- User can manually enter patient details vital to heart disease diagnosis.

2. View history

- The staff needs to login to access heart disease software.
- Staff can add, manage patient details.
- The staff can generate history of patient details using the website.

3. Database

- The user details are stored in Database.
- The database is accessed to get history of prediction details.
- The user details and their generated passwords are stored on the database.

4.9. SYSTEM DESIGN

Once the requirements document for the software to be developed is available, the software design phase begins. While the requirement specification activity deals entirely with the problem domain, design is the first phase of transforming the problem into a solution. In the design phase, the customer and business requirements and technical considerations all come together to formulate a product or a system.

The design process comprises a set of principles, concepts and practices, which allow a software engineer to model the system or product that is to be built. This model, known as design model, is assessed for quality and reviewed before a code is generated and tests are conducted. The design model provides details about software data structures, architecture, interfaces and components which are required to implement the system. This chapter discusses the design elements that are required to develop a software design model. It also discusses the design patterns and various software design notations used to represent a software design.

4.9.1. Data Flow Diagram

Data flow diagrams (DFDs) reveal relationships among and between the various components in a program or system. DFDs are an important technique for modeling a system's high-level detail by showing how input data is transformed to output results through a sequence of functional transformations. DFDs consist of four major components: entities, processes, data stores, and data flows. The symbols used to depict how these components interact in a system are simple and easy to understand; however, there are several DFD models to work from, each having its own symbology. DFD syntax does remain constant by using simple verb and noun constructs. Such a syntactical relationship of DFDs makes them ideal for object-oriented analysis and parsing functional specifications into precise DFDs for the systems analyst.

When it comes to conveying how information data flows through systems (and how that data is transformed in the process), data flow diagrams (DFDs) are the method of choice over technical descriptions for three principal reasons.

1. DFDs are easier to understand by technical and nontechnical audiences.

2. DFDs can provide a high level system overview, complete with boundaries and connections to other systems.

3. DFDs can provide a detailed representation of system components.

DFDs help system designers and others during initial analysis stages visualize a current system or one that may be necessary to meet new requirements. Systems analysts prefer working with DFDs, particularly when they require a clear understanding of the boundary between existing systems and postulated systems. DFDs represent the following:

1. External devices sending and receiving data.

2. Processes that change that data.

3. Data flows themselves.

4. Data storage locations.

The hierarchical DFD typically consists of a top-level diagram (Level 0) underlain by cascading lower level diagrams (Level 1, Level 2…) that represent different parts of the system.

External Entity

An external entity, which are also known as terminators, sources, sinks, or actors, are an outside system or process that sends or receives data to and from the diagrammed system. They're either the sources or destinations of information, so they're usually placed on the diagram's edges.

Process

Process is a procedure that manipulates the data and its flow by taking incoming data, changing it, and producing an output with it. A process can do this by performing computations and using logic to sort the data or change its flow of direction. Processes usually start from the top left of the DFD and finish on the bottom right of the diagram.

Data Store

Data stores hold information for later use, like a file of documents that's waiting to be processed. Data inputs flow through a process and then through a data store while data outputs flow out of a data store and then through a process.

Data flow

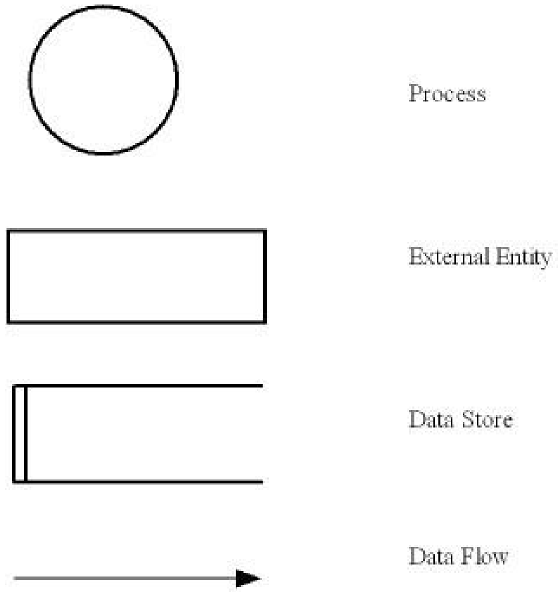


Fig 4.8 DFD symbols

Data flow is the path the system's information takes from external entities through processes and data stores. With arrows and succinct labels, the DFD can show you the direction of the data flow.

Level 0 DFDs, also known as context diagrams, are the most basic data flow diagrams. They provide a broad view that is easily digestible but offers little detail. Level 0 data flow diagrams show a single process node and its connections to external entities.

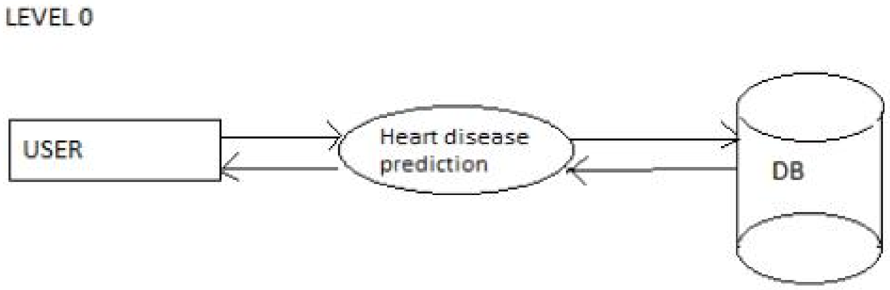


Fig 4.9 Level 0 DFD

Level 1 DFDs are still a general overview, but they go into more detail than a context diagram. In a level 1 data flow diagram, the single process node from the context diagram is broken down into subprocesses. As these processes are added, the diagram will need additional data flows and data stores to link them together.

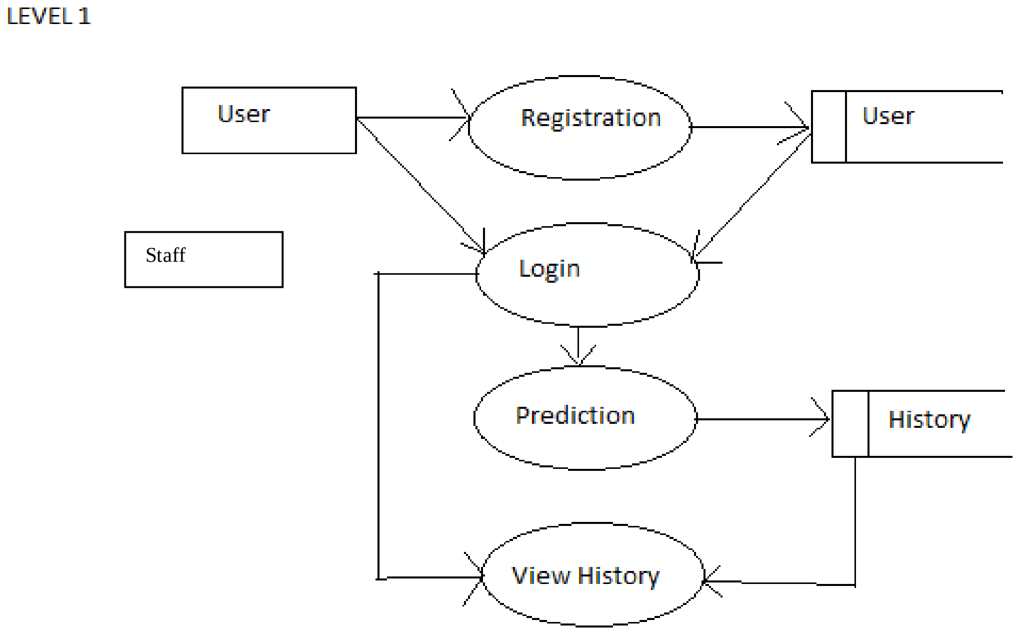


Fig 4.10 Level 1.1 DFD

4.9.2. Database Design

The database design is a collection of stored data organized in such a way that the data requirements are satisfied by the database, The general objective is to make information access easy, quick inexpensive and flexible for the user. There are also some specific objectives like controlled redundancy from failure, privacy, security, and performance. A collection of relative records make up a table. To design and store data to the needed forms database tables are prepared. Two essential settings for a database are:

- Primary key: The field that is unique for all the record occurrences.
- Foreign key: The field used to set relation between tables.
- Normalization is a technique to avoid redundancy in the tables.

Database design requires managing the large bodies of information. The management of data involves both the definition of structure of the storage of information and provision of mechanism for the manipulation of information. In addition the DB system must provide for the safety of information handled, despite the system crashes due to attempts art unauthorized access. For developing an efficient db, we will have to fulfil certain condition such as:

- **Control redundancy:** Redundancy is the duplication of critical components or functions of a system with the intention of a increasing reliability of the system. There are two types of redundancy; known as passive redundancy and active redundancy. Both functions prevent performance decline from exceeding specification limits without human intervention using extra capacity.
- **Data independence:** Data independency is the type of data transparency that matters for a centralized DBMS. It refers to the immunity of user applications to changes made in the definition and organization of data. The physical data independence deals with hiding the details of the storage structure from user applications. The data independence and operation independence together gives the feature of data abstraction.

- **Accuracy and integrity:** Data integrity refers to maintaining and assuring the accuracy and consistency of the data over its entire life cycle, and is a critical aspect to the design, implementation, and usage of any system which stores, processes, or retrieves the data. Accuracy refers to getting the extracted data.
- **Avoiding inordinate delays:** Difficulties in managing DBMS delays managing performance are more challenging. DBMS stores information on hard disks. This has different types of delays.
- **Ease of use:** It is also referred to as usability. It is the ease of use and learn ability of a human made object. It also includes methods of measuring usability, such as needs analysis and the study of the principles behind an object's perceived efficiency or elegance.
- **Recovery from failure:** In DBMS failures are generalized into various categories. They are transaction failures, system crash, disk failures etc. When a DBMS recovers from a crash, it should maintain some characteristics. It should check the states of all the transactions, which were being executed, it should check whether the transaction can be completed now or it needs to be rolled back etc.
- **Private and security:** Database security concerns the use of a broad range of information security controls to protect databases against compromises of their confidentiality, integrity and availability. Security risks to database system include unauthorized or unintended activity or misuse by authorized database users, database administrators or network/systems managers or by unauthorized users or hackers etc.

List of tables used in this project:

Login, ops, user

Login

Lid	UserName	Password	type

Ops

Pid	Age	cp	Trestbps	Chol	fbs	restecg	thalach	exang	oldpeak	Slope	Ca	thal	gender	result	lid

User

User_id	l_id	Name	Dob	Place	dist	phone	gender	email

Fig

CHAPTER 5

SYSTEM IMPLEMENTATION AND TESTING

5.1. SOFTWARE REQUIREMENT

The proposed project is developed using:

Processor	: Intel i5^{8th} generation
Processor Speed	: 3.2 GHz
Main storage	: 8GB RAM
Hard disk capacity	: 1 TB
Keyboard	: 104 keys
Monitor	: 15.6 FHD
Mouse	: Logitech

5.1.1. Front end

In this project html, css and Android XML are used

(1)html

HTML stands for Hypertext Markup Language, and it is the most widely used language to write Web Pages.

• Hypertext refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a webpage is called Hypertext.

• As its name suggests, HTML is a Markup Language which means you use HTML to simply "mark-up" a text document with tags that tell a Web browser how to structure it to display.

Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers. Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out what background images or colors are used, layout designs,variations in display for different devices and screen sizes as well as a variety of other effects.CSS is easy to learn and understand but it provides powerful control over the presentation of

an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

(2)Android XML

XML stands for Extensible Mark-up Language.XML is a very popular format and commonly used for sharing data on the internet. This chapter explains how to parse the XML file and extract necessary information from it.

Android provides three types of XML parsers which are DOM,SAX and XMLPullParser. Among all of them android recommend XMLPullParser because it is efficient and easy to use. So we are going to use XMLPullParser for parsing XML.

The first step is to identify the fields in the XML data in which you are interested in.

XML - Elements

An xml file consists of many components. Here are the definition of components of an XML file and their description.

1. Prolog

An XML file starts with a prolog. The first line that contains the information about a file is prolog

2. Events

An XML file has many events. Event could be like this. Document starts , Document ends, Tag start , Tag end and Text etc.

3. Text

Apart from tags and events, and xml file also contains simple text. Such as GB is a text in the country tag.

4. Attributes

Attributes are the additional properties of a tag such as value e.t.c

XML - Parsing

In the next step, we will create XMLPullParser object , but in order to create that we will first create XmlPullParserFactory object and then call its newPullParser() method to create XMLPullParser. Its syntax is given below −

```
 XmlPullParserFactory xmlFactoryObject = XmlPullParserFactory.newInstance();
private XmlPullParser myparser = xmlFactoryObject.newPullParser();
```

The next step involves specifying the file for XmlPullParser that contains XML. It could be a file or could be a Stream. In our case it is a stream.Its syntax is given below −

```
myparser.setInput(stream, null);
```

The last step is to parse the XML. An XML file consist of events, Name, Text, AttributesValue e.t.c. So XMLPullParser has a separate function for parsing each of the component of XML file.

(3) CSS

CSS stands for Cascading Style Sheets

CSS describes how HTML elements are to be displayed on screen, paper, or in other media

CSS saves a lot of work. It can control the layout of multiple web pages all at once

External stylesheets are stored in CSS files

5.1.2. Back end

In this project python and MySQL are used

1) Python

Python is a popular programming language.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

Python can be used on a server to create web applications.
Python can be used alongside software to create workflows.
Python can connect to database systems. It can also read and modify files.
Python can be used to handle big data and perform complex mathematics.
Python can be used for rapid prototyping, or for production-ready software development

5.1.2.1. MySQL

MySQL is an Oracle-backed open source relational database management system (RDBMS) based on Structured Query Language (SQL). MySQL runs on virtually all platforms, including Linux, UNIX and Windows. Although it can be used in a wide range of applications, MySQL is most often associated with web applications and online publishing. MySQL is an important component of an open source enterprise stack called LAMP. LAMP is a web development platform that uses Linux as the operating system, Apache as the web server, MySQL as the relational database management system and PHP as the object-oriented
scripting language. (Sometimes Perl or Python is used instead of PHP).
Features:

- Relational Database Management System (RDBMS): MySQL is a relational database management system.

- Easy to use: MySQL is easy to use. You have to get only the basic knowledge of SQL. You can build and interact with MySQL with only a few simple SQL statements.
- It is secure: MySQL consist of a solid data security layer that protects sensitive data from intruders. Passwords are encrypted in MySQL.
- Client/ Server Architecture: MySQL follows a client /server architecture. There is a database server (MySQL) and arbitrarily many clients (application programs), which communicate with the server; that is, they query data, save changes, etc.
- Free to download: MySQL is free to use and you can download it from MySQL official website.
- It is scalable: MySQL can handle almost any amount of data, up to as much as 50 million rows or more. The default file size limit is about 4 GB. However, you can increase this number to a theoretical limit of 8 TB of data.
- Compatible on many operating systems: MySQL is compatible to run on many
- operating systems, like Novell NetWare, Windows* Linux*, many varieties of
- UNIX* (such as Sun* Solaris*, AIX, and DEC* UNIX), OS/2, FreeBSD*, and others.
- MySQL also provides a facility that the clients can run on the same computer as the server or on another computer (communication via a local network or the Internet).
- Allows roll-back: MySQL allows transactions to be rolled back, commit and crash recovery.
- High Performance: MySQL is faster, more reliable and cheaper because of its unique storage engine architecture.
- High Flexibility: MySQL supports a large number of embedded applications which makes MySQL very flexible.
- High Productivity: MySQL uses Triggers, Stored procedures and views which allows the developer to give a higher productivity.

5.1.2.2. PyCharm IDE

PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django as well as Data Science with Anaconda. PyCharm is cross platform, with Windows, macOS and Linux versions.

FEATURES:

- Coding assistance and analysis, with code completion, syntax and error highlighting, linter integration, and quick fixes
- Project and code navigation: specialized project views, file structure views and quick jumping between files, classes, methods and usages
- Python refactoring: includes rename, extract method, introduce variable, introduce constant, pull up, push down and others
- Support for web frameworks: Django, web2py and Flask [professional edition only] [9]
- Integrated Python debugger
- Integrated unit testing, with line-by-line code coverage
- Google App Engine Python development [professional edition only

Support for scientific tools like matplotlib, numpy and scipy [professional edition only.

For developing the Android application “Android studio” and for working with database

MySql “SQLyog” tool is used. Due to its simple UI, It is very easy to built and manage database using queries.

5.1.2.3 SQLyog

MySQL manager and admin tool:

SQLyog is the most powerful manager, admin and GUI tool for MySQL, combining the features of MySQL Query Browser, Administrator, phpMyAdmin and other MySQ Front Ends and MySQL GUI tools in a single intuitive interface. SQLyog is a fast, easy to use and compact graphical tool for managing your MySQL databases. SQLyog was developed for all who use MySQL as their preferred RDBMS. Whether you enjoy the control of handwritten

SQL or prefer to work in a visual environment, SQLyog makes it easy for you to get started and provides you with tools to enhance your MySQL experience.

□ Session Restore:

SQLyog is a powerful MySQL manager that restores your previous session the way you left it. Be it a system crash or accidentally closing your MySQL client.

□ Schema and Data Sync:

Find and fix schema mismatching while syncing data from one database to another. Replicate data periodically to a different server at scheduled intervals.

□ Compressed backups with scheduling:

Find the plethora of options in mysqldump daunting? Just back up with a single click interface in SQLyog MySQL GUI tool.

□ Auto complete and SQL formatting:

Save hours of typing. Write queries 10x faster. Inherited spaghetti SQL? Make it neat with a click. Improve readability and maintenance of SQL.

□ Query Builder:

Build complex queries using drag-n-drop interface. Visually create SQL statements without the need to remember column names.

□ Schema Optimizer and Index Analyzer:

Determine the optimal column types. Fit more data into memory thereby improving response times. Eliminate Redundant Indexes by improving INSERTs and UPDATEs.

□ Object Browser:

The Object browser window gives you complete details of the server you are connected to in a tree format. It allows you to perform different operations on selected objects such as Open Table, Create Table, Alter Table, Manage Indexes, Relationships/Foreign Keys, Backup/Export, Import and more.

□ Favorites :

SQLyog allows easy access to frequently used SQL scripts. The script files can be stored as files from the SQLyog interface or link to an existing file anywhere where windows can access it - on a local drive or a shared network drive. The SQL scripts and the file links can be organized in folder and subfolders. SQLyog gives you a choice between four different connectivity options. No matter whatever your firewall/proxy/ISP settings are, you should be able to smartly manage your MySQL Server(s). The various connectivity options (in the order of responsiveness and network efficiency) are:

□ Direct connection using MySQL C API: This is by far the fastest way to connect and manage your MySQL server. This is the preferred method if you are not concerned about your MySQL traffic being spoofed. However, this requires direct access to your MySQL server. Most ISPs block the MySQL port or disallow remote access.

□ SSH Tunneling: Use this option if you need to encrypt the contents of your MySQL traffic, but still want the efficiency of the MySQL C API. If your host supports SSH tunneling, SQLyog can use the tunnel to communicate with MySQL.

□ HTTP Tunneling: This option saves your day if your ISP/MySQL provider disallows remote access, either by blocking the MySQL port or by giving access to “local host” only. The response is slow compared to the previous options since the data is XML encoded and HTTP is stateless by nature. However, you can use all the cool features of SQLyog.

□ HTTPS Tunneling: This option is similar to HTTP Tunneling but adds another layer of security by establishing a secure channel.

□ SSL Encryption: With this option all communication between SQLyog and the MySQL server is encrypted. The MySQL server must support this option. Not all servers do.

5.1.2.4 Android Studio IDE

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA. On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building

Android apps, such as:

□ A flexible Gradle-based build system.

□ A fast and feature-rich emulator.

□ A unified environment where you can develop for all Android devices.

□ Instant Run to push changes to your running app without building a new APK.

□ Code templates and GitHub integration to help you build common app features and import sample code.

□ Lint tools to catch performance, usability, version compatibility, and other problems.

□ C++ and NDK support.

□ Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine.

□

Android features:

1. Instant App Run

It is an advanced technology in which it cleverly understands the transmutations done in the applications and deliver it instantly without taking time to rebuild the apk and installations make. So, quick that you can see the changes in app immediately. This is done by launching the URL on Android application installing the native libraries with Android Instant apps.

Instant App sometimes is known as " New Module Wizard".

2. Visual Layout Editor

Layout editor helps to build the layout quickly by adding different attributes either by hard-code or drag and drop. The preview of the codes can be seen easily on the visual editor screen

and changes can be made accordingly by resizing it dynamically. This will make testing the application process more facile and more exhaustive.

3. Fast Emulator

Android has a great feature of Emulator which is exactly like the android phones to test how the application looks like in physical devices. It gives real-time experience to the Android applications. It allows you to test your applications faster and on different-different configuration devices like tablet, android phone etc. It helps you to make your application development life cycle shorter and more efficient.

4. Intelligence Code Editor

Android Studio provides you with the intelligent and quick code editor. This will help you and guide you with the accurate code. It helps you to complete code in advance and analyse your code in advance before building. Android studio has the special feature of the development of code by the drop-down list with suggesting the code you can integrate.

5. Addition of New Activity as a Code Template

Yes, Android also has the feature of templates built-in. If you know to build that accordingly that makes your task easier. it has both pros and cons, you don't find every template in

Android Studio. It's an additional feature which helps the developer to build an application efficiently and effectively which provide effective solutions.

6. Help to Build Up App for All Devices

Android studio builds applications for every screen size, for wear and gear devices etc. It also can stimulate the various type of features which a hardware has like GPS location tracker, multi-touch.

7. Help to Connect with Firebase

Android Studio helps to give real-time experience with IOT based project development with dynamic upgrades in the application. Firebase connectivity help to create direct updates and provide databases connectivity. To build high-quality applications we must use Firebase connectivity it helps to build the scalable infrastructure for building the application. You can

create chat applications by using firebase connectivity it helps you to do happy chat experience.

8.Support KOTLIN

Kotlin the official language can be considered for Android. It is a language without having any new restrictions and has various advantages in it. The great feature of Kotlin is it run without any disturbances in older android versions as well that means no issues of specific android versions. It runs fast and equivalent to Java. Java developers can easily hands-on

Kotlin with no-issues because it based on automated Java only.

9. Color Previews

Android studio helps to see the code XML part in a preview to know that how perfectly we are designing the application according to the need before launching the application. It

provides powerful functionality and enhanced features of drag and drops or resizes the application. It contains drag and drop features but not support for every function, that's why be careful while doing that.

10. Maven Repository

In Android Studio, Maven integration of its repository can be done, within SDK manager support libraries of IDE is used. It’s a kind of a repository which is a directory in which various jar files like project jars, Plugin are stored.

5.2 HARDWARE REQUIREMENTS

Processor: Intel Core i3

Processor Speed: 2.4GHZ

RAM: 4GB

5.3 INSTALLATION REQUIREMENTS

Smart Phone with Android Operating System

Android version: 4.4 and above (i.e., kitkat and above)

5.3.1 Android Operating System

Android is an open source and Linux based operating system for mobile devices such as

Smartphone and tablet computers. Android was developed by the open handset Alliance, led by Google, and other companies. Google wanted Android to be open and free; hence most of the Android code was released under the open source Apache license, which means that

anyone who wants to use Android can do so by downloading the full Android source code.

vendors (i.e. hardware manufactures) can add their own proprietary extensions to Android

and customize Android to differentiate their product from others. The main advantage of adopting Android is that it offers a unified approach to application development. developer need only develop for Android, and their application should be able to run numerous different devices as long as these devices are powered using android. Application is the most important part of success chain in the world of Android.

Features of Android

Android is open source and freely available to manufacturers for customization therefore there are no fixed hardware or software configurations. However Android itself support the following features;

 Storage – Uses MySQL for data storage

Connectivity - Support GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth(includes A2DP and AVRCP), Wi-Fi, LTE and WiMAX

Messaging - Supports both SMS and MMS

Web browser - Based on open source web kit together with chrome’s Vs. JavaScript engine

Media Support- Includes support for the following media: H.263, H.264 (in 3GP 3GP or MP4 container), MPEG-4 SP, AMR-WB(in 3GP container), AAC, HE-AAC(in MP4 or 3GP container), MP3, MIDI, WAV, JPEG, PNG, GIF and BMP

Hardware Support - Accelerometer sensor, Camera, Digital Compass, Proximity sensor and GPS

Multi-touch - Supports multi-touch screens Multi-tasking - Support multi-tasking applications

Flash Support - Android 2.3 supports flash 10.1

Tethering - supports sharing of internet connection as a wired/wireless Hotspot

The main Android devices in the market include Smartphone, Tablet, E-reader devices, net books, MP4 players, internet TVs and so on.

CHAPTER 6

RESULT ANALYSIS AND DISCUSSION

An enhanced heart disease prediction system is successfully built with all the specified functionalities, giving the expected outcome at each step, the data successfully flow between client and server without problems. The proposed hybrid HRFLM approach is used combining the characteristics of Random Forest (RF) and Linear Method (LM). HRFLM proved to be quite accurate in the prediction of heart disease.

The system also performs the task with a good performance, an addition it's easily used requiring uncomplicated configurations to be used, so it's a user-friendly application. The developed system produce an enhanced performance level with an accuracy level of 88.7% through the prediction model for heart disease with the hybrid random forest with a linear model (HRFLM).

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

The proposed system is GUI-based, user-friendly, scalable, reliable and an expandable system. The proposed working model can also help in reducing treatment costs by providing Initial diagnostics in time. The model can also serve the purpose of training tool for medical students and will be a soft diagnostic tool available for physician and cardiologist. General physicians can utilize this tool for initial diagnosis of cardio-patients. There are many possible improvements that could be explored to improve the scalability and accuracy of this prediction system. As we have developed a generalized system, in future we can use this system for the analysis of different data sets. The performance of the health's diagnosis can be improved significantly by handling numerous class labels in the prediction process, and it can be another positive direction of research. In machine learning, the identification and selection of significant attributes for better diagnosis of heart disease are very challenging tasks for future research.

REFERENCES

1. R. Das, I. Turkoglu, and A. Sengur, Effective diagnosis of heart disease through neural networks ensembles, Expert Syst. Appl., vol. 36, no. 4, pp. 7675–7680, May 2009

2. D. K. Ravish, K. J. Shanthi, N. R. Shenoy, and S. Nisargh, ‘‘Heart function monitoring, prediction and prevention of heart attacks: Using artificial neural networks,’’ in Proc. Int. Conf. Contemp. Comput. Inform. (IC3I),
Nov. 2014, pp. 1–6.

3. M. Durairaj and V. Revathi, ‘‘Prediction of heart disease using back propagation MLP algorithm,’’ Int. J. Sci. Technol. Res., vol. 4, no. 8, pp. 235–239, 2015.

4. O. W. Samuel, G. M. Asogbon, A. K. Sangaiah, P. Fang, and G. Li,
‘‘An integrated decision support system based on ANN and Fuzzy_AHP for heart failure risk prediction,’’ Expert Syst. Appl., vol. 68, pp. 163–172, Feb. 2017.

5. A. Lakshmanarao,Y.Swathi, P.Sri Sai Sundareswar, “Machine learning techniques for heart disease prediction” Int. J. Sci. Technol. Res., vol 4, issue 11, november 2019

6. V. Krishnaiah, G. Narsimha, and N. Subhash, ‘‘Heart disease prediction system using data mining techniques and intelligent fuzzyapproach: A review,’’ Int. J. Comput. Appl., vol. 136, no. 2, pp. 43–51,
2016.

7. C. Sowmiya and P. Sumitra, ‘‘Analytical study of heart disease diagnosis using classification techniques,’’ in Proc. IEEE Int. Conf. Intell. Techn. Control, Optim. Signal Process. (INCOS), Mar. 2017.

APPENDIX A

hrlm.py – Heart Disease Predictor

```
from flask import Flask,render_template,request,jsonify,redirect
from dbconnection import Db

app = Flask(__name__)



@app.route('/')
def a():
    return render_template('login.html')
@app.route('/admhome')
def admhome():
    return render_template('admhome.html')

@app.route('/admusers')
def admusers():
    db=Db()
    qry="SELECT * FROM USER"
    res=db.select(qry)
    return render_template('admsers.html',data=res)

@app.route('/more/<lid>')
def more(lid):
    db=Db()
    qry="SELECT * FROM ops WHERE lid='"+lid+"'"
    res=db.select(qry)
    return render_template('admops.html',data=res)



@app.route('/lik')
def lik():
    import numpy as np
    from sklearn.ensemble import RandomForestClassifier
    from sklearn import datasets
    from sklearn.model_selection import train_test_split
    from sklearn.feature_selection import SelectFromModel
    from sklearn.metrics import accuracy_score

    import numpy as np
    import matplotlib.pyplot as plt
    import pandas as pd

    # Importing the datasets
    feat_labels = ['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach', 'exang', 'oldpeak', 'slope',
                   'ca', 'thal']

    datasets = pd.read_csv(r'D:\Backups\hrflm\hrflm\hrflm\static\heart.xls')
    X = datasets.iloc[:, 0:13].values
    Y = datasets.iloc[:, 13].values

    X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.25, random_state=0)

    clf = RandomForestClassifier(n_estimators=10000, random_state=0, n_jobs=-1)

    # lk Train the classifier
```

```
clf.fit(X_train, y_train)

# lk Print the name and gini importance of each feature

aa=[]

for feature in zip(feat_labels, clf.feature_importances_):
    print(feature)
    aa.append(feature)

sfm = SelectFromModel(clf, threshold=0.12)

sfm.fit(X_train, y_train)

X_important_train = sfm.transform(X_train)
X_important_test = sfm.transform(X_test)

# print(X_important_test)

#
# clf_important = RandomForestClassifier(n_estimators=10000, random_state=0, n_jobs=-1)
# #
# # # Train the new classifier on the new dataset containing the most important features
# clf_important.fit(X_important_train, y_train)

from sklearn.linear_model import LogisticRegression
clf_important=LogisticRegression()
clf_important.fit(X_important_train, y_train)
```

```
    # print("trained")
    #
    y_pred = clf_important.predict(X_important_test)
    print("predicted")
    # # View The Accuracy Of Our Full Feature (4 Features) Model
    s=accuracy_score(y_test, y_pred)
    #
    print(s)
    print("ok")

    #######################


    from sklearn.metrics import confusion_matrix
    cm = confusion_matrix(y_test, y_pred)

    print(cm)

    from sklearn.metrics import roc_auc_score, average_precision_score, roc_curve
    fpr, tpr, thresholds = roc_curve(y_test, y_pred)
    rocval = roc_auc_score(y_test, y_pred)

    avc = average_precision_score(y_test, y_pred)

    # Visualising the Training set results

    return render_template('confusionmatrix.html', cm=cm, rocval=rocval,
average_precision_score=avc, fpr=fpr,
                    tpr=tpr,aa=aa)
```

```
@app.route('/likrf')
def likknn():
    import numpy as np
    from sklearn.ensemble import RandomForestClassifier
    from sklearn import datasets
    from sklearn.model_selection import train_test_split
    from sklearn.feature_selection import SelectFromModel
    from sklearn.metrics import accuracy_score

    import numpy as np
    import matplotlib.pyplot as plt
    import pandas as pd

    # Importing the datasets
    feat_labels = ['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach', 'exang',
'oldpeak', 'slope',
                   'ca', 'thal']

    datasets = pd.read_csv(r'D:\Backups\hrflm\hrflm\hrflm\static\heart.xls')
    X = datasets.iloc[:, 0:13].values
    Y = datasets.iloc[:, 13].values

    X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.25, random_state=0)

    clf = RandomForestClassifier(n_estimators=10000, random_state=0, n_jobs=-1)

    # lk Train the classifier
    clf.fit(X_train, y_train)

    # lk Print the name and gini importance of each feature
    y_pred=clf.predict(X_test)

    s=accuracy_score(y_test, y_pred)
    #
    print(s)
    print("ok")

    ########################

    from sklearn.metrics import confusion_matrix
    cm = confusion_matrix(y_test, y_pred)

    print(cm)

    from sklearn.metrics import roc_auc_score, average_precision_score, roc_curve
    fpr, tpr, thresholds = roc_curve(y_test, y_pred)
    rocval = roc_auc_score(y_test, y_pred)

    avc = average_precision_score(y_test, y_pred)

    # Visualising the Training set results

    return render_template('confusionmatrixrf.html', cm=cm, rocval=rocval,
average_precision_score=avc, fpr=fpr,
                  tpr=tpr)
```

```
@app.route('/liknb')
def liknb():
    import numpy as np
    from sklearn.naive_bayes import GaussianNB
    from sklearn import datasets
    from sklearn.model_selection import train_test_split
    from sklearn.feature_selection import SelectFromModel
    from sklearn.metrics import accuracy_score

    import numpy as np
    import matplotlib.pyplot as plt
    import pandas as pd

    # Importing the datasets
    feat_labels = ['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach', 'exang',
'oldpeak', 'slope',
                   'ca', 'thal']

    datasets = pd.read_csv(r'D:\Backups\hrflm\hrflm\hrflm\static\heart.xls')
    X = datasets.iloc[:, 0:13].values
    Y = datasets.iloc[:, 13].values

    X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.25, random_state=0)

    clf = GaussianNB()

    # lk Train the classifier
    clf.fit(X_train, y_train)

    # lk Print the name and gini importance of each feature
    y_pred=clf.predict(X_test)
    s=accuracy_score(y_test, y_pred)
    #
    print(s)
    print("ok")

    ########################

    from sklearn.metrics import confusion_matrix
    cm = confusion_matrix(y_test, y_pred)

    print(cm)

    from sklearn.metrics import roc_auc_score, average_precision_score, roc_curve
    fpr, tpr, thresholds = roc_curve(y_test, y_pred)
    rocval = roc_auc_score(y_test, y_pred)

    avc = average_precision_score(y_test, y_pred)

    # Visualising the Training set results

    return render_template('confusionmatrixrf.html', cm=cm, rocval=rocval,
average_precision_score=avc, fpr=fpr,
                           tpr=tpr)




@app.route('/likdt')
def likdt():
    import numpy as np
    from sklearn.tree import DecisionTreeClassifier
    from sklearn import datasets
    from sklearn.model_selection import train_test_split
    from sklearn.feature_selection import SelectFromModel
    from sklearn.metrics import accuracy_score

    import numpy as np
    import matplotlib.pyplot as plt
    import pandas as pd

    # Importing the datasets
    feat_labels = ['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach', 'exang',
'oldpeak', 'slope',
               'ca', 'thal']

    datasets = pd.read_csv(r'D:\Backups\hrflm\hrflm\hrflm\static\heart.xls')
    X = datasets.iloc[:, 0:13].values
    Y = datasets.iloc[:, 13].values

    X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.25, random_state=0)

    clf = DecisionTreeClassifier()

    # lk Train the classifier
    clf.fit(X_train, y_train)

    # lk Print the name and gini importance of each feature
    y_pred=clf.predict(X_test)

    s=accuracy_score(y_test, y_pred)
    #
    print(s)
```

```
    print("ok")

    ########################


    from sklearn.metrics import confusion_matrix
    cm = confusion_matrix(y_test, y_pred)

    print(cm)

    from sklearn.metrics import roc_auc_score, average_precision_score, roc_curve
    fpr, tpr, thresholds = roc_curve(y_test, y_pred)
    rocval = roc_auc_score(y_test, y_pred)

    avc = average_precision_score(y_test, y_pred)

    # Visualising the Training set results

    return render_template('confusionmatrixrf.html', cm=cm, rocval=rocval,
average_precision_score=avc, fpr=fpr,
                           tpr=tpr)






@app.route('/liklr')
def liklr():
    import numpy as np
    from sklearn.linear_model import LinearRegression
    from sklearn import datasets
    from sklearn.model_selection import train_test_split
    from sklearn.feature_selection import SelectFromModel
    from sklearn.metrics import accuracy_score

    import numpy as np
    import matplotlib.pyplot as plt
    import pandas as pd

    # Importing the datasets
    feat_labels = ['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach', 'exang',
'oldpeak', 'slope',
               'ca', 'thal']

    datasets = pd.read_csv(r'D:\Backups\hrflm\hrflm\hrflm\static\heart.xls')
    X = datasets.iloc[:, 0:13].values
    Y = datasets.iloc[:, 13].values

    X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.25, random_state=0)

    clf = LinearRegression()

    # lk Train the classifier
    clf.fit(X_train, y_train)

    # lk Print the name and gini importance of each feature
    y_pred=clf.predict(X_test)

    s=accuracy_score(y_test, y_pred)
    #
    print(s)
    print("ok")
```

```
    #######################


    from sklearn.metrics import confusion_matrix
    cm = confusion_matrix(y_test, y_pred)

    print(cm)

    from sklearn.metrics import roc_auc_score, average_precision_score, roc_curve
    fpr, tpr, thresholds = roc_curve(y_test, y_pred)
    rocval = roc_auc_score(y_test, y_pred)

    avc = average_precision_score(y_test, y_pred)

    # Visualising the Training set results

    return render_template('confusionmatrixrf.html', cm=cm, rocval=rocval,
average_precision_score=avc, fpr=fpr,
                           tpr=tpr)




@app.route('/liksvm')
def liksvm():
    import numpy as np
    from sklearn.svm import SVC
    from sklearn import datasets
    from sklearn.model_selection import train_test_split
    from sklearn.feature_selection import SelectFromModel
    from sklearn.metrics import accuracy_score
```

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the datasets
feat_labels = ['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach', 'exang',
'oldpeak', 'slope',
               'ca', 'thal']

datasets = pd.read_csv(r'D:\Backups\hrflm\hrflm\hrflm\static\heart.xls')
X = datasets.iloc[:, 0:13].values
Y = datasets.iloc[:, 13].values

X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.25, random_state=0)

clf = SVC(gamma='auto')


# lk Train the classifier
clf.fit(X_train, y_train)

# lk Print the name and gini importance of each feature
y_pred=clf.predict(X_test)

s=accuracy_score(y_test, y_pred)
#
print(s)
print("ok")

########################
```

```
    from sklearn.metrics import confusion_matrix
    cm = confusion_matrix(y_test, y_pred)

    print(cm)

    from sklearn.metrics import roc_auc_score, average_precision_score, roc_curve
    fpr, tpr, thresholds = roc_curve(y_test, y_pred)
    rocval = roc_auc_score(y_test, y_pred)

    avc = average_precision_score(y_test, y_pred)

    # Visualising the Training set results

    return render_template('confusionmatrixrf.html', cm=cm, rocval=rocval,
average_precision_score=avc, fpr=fpr,
                           tpr=tpr)




@app.route('/applogin',methods=['post'])
def applogin():
    username=request.form["uname"]
    password=request.form["password"]
    db=Db()

    qry="SELECT * FROM login WHERE `user_name`='"+username+"' AND
PASSWORD='"+password+"'"
    res=db.selectOne(qry)

    if res is not None:
        if res['type']=="admin":
            return redirect('/admhome')
        else:
            return "<script>alert('No privilege to access'); window.location='/'</script>"
    else:
        return "<script>alert('Invalid username or password');
window.location='/'</script>"




@app.route('/and_signup',methods=['post'])
def and_signup():

    db=Db()

    name=request.form["uname"]
    dob=request.form["dob"]
    place=request.form["place"]
    dist=request.form["dist"]
    phone=request.form["phone"]
    gender=request.form["gender"]
    email=request.form["email"]
    passw=request.form["pass"]

    qry="INSERT INTO `login` (`user_name`,`password`,`type`) VALUES
('"+email+"','"+passw+"','user')"
    lid=str(db.insert(qry))

    print(lid)
    qry="INSERT INTO `user`
(`l_id`,`name`,`dob`,`place`,`dist`,`phone`,`gender`,`email`)  VALUES
('"+lid+"','"+name+"','"+dob+"','"+place+"','"+dist+"','"+phone+"','"+gender+"','"+e
mail+"')"
    print(qry)

    db.insert(qry)

    return jsonify(status="ok")



@app.route('/and_login',methods=['post'])
def and_login():
    db=Db()
    uname=request.form["uname"]
    pwd=request.form["pwd"]
    qry="SELECT * FROM login WHERE user_name='"+uname+"' AND
PASSWORD='"+pwd+"'"
    res=db.selectOne(qry)

    if res is not None:
        return jsonify(status='ok',uid=res['l_id'])
    else:
        return jsonify(status='no')

@app.route('/and_changepassword',methods=['post'])
def and_changepassword():
    opass=request.form["opass"]
    npass=request.form["npass"]
    lid=request.form["lid"]
    db=Db()
    qry="SELECT * FROM login WHERE l_id='"+lid+"' AND PASSWORD='"+opass+"'"
    res=db.selectOne(qry)
    if res is None:
        return jsonify(status='no')
    else:
        qry="UPDATE login SET PASSWORD='"+npass+"' WHERE l_id='"+lid+"'"
        db.insert(qry)
        return jsonify(status='ok')

@app.route('/and_history',methods=['post'])
def and_history():
    lid=request.form["lid"]
    qry="SELECT * FROM ops WHERE lid='"+str(lid)+"'"
    db=Db()
    res=db.select(qry)
    return jsonify(status="ok",data=res)



@app.route('/and_predict', methods=['post'])
def and_predict():
    age = float(request.form["age"])
    cp = float(request.form["cp"])
    trestbps = float(request.form["trestbps"])
    chol = float(request.form["chol"])

    fbs = float(request.form["fbs"])
    restecg = float(request.form["restecg"])
    thalach = float(request.form["thalach"])
    exang = float(request.form["exang"])
    oldpeak = float(request.form["oldpeak"])
    slope = float(request.form["slope"])
    ca = float(request.form["ca"])
    thal = float(request.form["thal"])

    gender = float(request.form["gender"])

    test = [[age, gender, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, ca,
thal]]

    print("lllllllllllllllll")

    import numpy as np
    from sklearn.ensemble import RandomForestClassifier
    from sklearn import datasets
    from sklearn.model_selection import train_test_split
    from sklearn.feature_selection import SelectFromModel
    from sklearn.metrics import accuracy_score

    import numpy as np
    import matplotlib.pyplot as plt
    import pandas as pd

    # Importing the datasets
    feat_labels = ['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach', 'exang',
'oldpeak', 'slope',
                   'ca', 'thal']

    print("mmmmmmmmmmmmm")
```

```
datasets = pd.read_csv(r'D:\Backups\hrflm\hrflm\hrflm\static\heart.xls')

print("aaaaaaaaaaaaa")

X = datasets.iloc[:, 0:13].values
Y = datasets.iloc[:, 13].values

# print("ms-------------------------")
#
#
#
# X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.25, random_state=0)

clf = RandomForestClassifier()

# lk Train the classifier
clf.fit(X, Y)
print("aaaaaaaaaaaaaaabbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbccc")
# lk Print the name and gini importance of each feature

#
#
# sfm = SelectFromModel(clf, threshold=0.0)
#
# sfm.fit(X_train, y_train)
#
# return jsonify(status="no")
#
# X_important_train = sfm.transform(X_train)
# X_important_test = sfm.transform(X_test)
#
```

```
    # from sklearn.linear_model import LogisticRegression
    # clf_important=LogisticRegression()
    # clf_important.fit(X_important_train, y_train)
    #
    res = clf.predict(test)
    print("lk1")
    lid = request.form["lid"]
    print("lk2")
    print(res)

    qry = "INSERT INTO `ops`
(`age`,`cp`,`trestbps`,`chol`,`fbs`,`restecg`,`thalach`,`exang`,`oldpeak`,`slope`,`ca`,`thal`,`
gender`,`result`,`lid`) VALUES ('" + str(
        age) + "','" + str(cp) + "','" + str(trestbps) + "','" + str(chol) + "','" + str(fbs) + "','" +
str(
        restecg) + "','" + str(thalach) + "','" + str(exang) + "','" + str(oldpeak) + "','" +
str(slope) + "','" + str(
        ca) + "','" + str(thal) + "','" + str(gender) + "','" + str(res[0]) + "','" + str(lid) + "')"
    db = Db()
    db.insert(qry)

    return jsonify(status='ok', stage=str(res[0]))



@app.route('/')
def hello_world():
    return render_template("index.html")

# @app.route('/and_history',methods=['post'])
# def and_history():
#     db=Db()
#     lid=str(request.form["lid"])
#     qry="SELECT * FROM ops WHERE lid='"+lid+"'"
#     res=db.select(qry)
#     return jsonify(status="ok",data=res)

@app.route('/admin_login',methods=["post"])
def admin_login():
    db=Db()
    uname=request.form["textfield"]
    password=request.form["textfield2"]
    qry="select *from login where user_name='"+uname+"' and
password='"+password+"'"
    res=db.selectOne(qry)
    if res is not None:
        return render_template("admin_home.html")
    else:
        return 'invalid user name or password'




@app.route('/alogin')
def alogin():
    return render_template("index.html")

@app.route('/comparison')
def comparison():
    return render_template("COMPARISON.html")

@app.route('/patientinfo')
def patientinfo():
    db=Db()
    qry="select *,floor(datediff(curdate(),dob)/365) as age from user"
    res=db.select(qry)
    return render_template("patient_info.html",data=res)

@app.route('/admin_home')
def admin_home():
    return render_template("admin_home.html")

@app.route('/logout')
def logout():
    return render_template("admin_ login.html")




if __name__ == '__main__':
    app.run(host='0.0.0.0',port=5000,debug=True)
```

RandomForestClassification_check.py

```
import pandas as pd
from sklearn.ensemble import RandomForestClassifier

balance_data=pd.read_csv('C:/Users/Admin/PycharmProjects/hrflm/dataset.csv',sep=',',
header=None)

X = balance_data.values[1:, 0:13]
Y = balance_data.values[1:, 13]




test_data=[[58,1,1,125,220,0,1,144,0,0.4,1,4,3]]
```

```
rfc = RandomForestClassifier()
rfc.fit(X,Y)
rfc_predict = rfc.predict(test_data)

print("prediction")
print(rfc_predict)
```

SampleKNN.py

```
import pandas as pd
from sklearn.neighbors import KNeighborsClassifier

balance_data=pd.read_csv('C:/Users/Admin/PycharmProjects/hrflm/dataset.csv',sep=',', header=None)

X= balance_data.values[1:, 0:13]
Y = balance_data.values[1:, 13]

test_data=[[41,1,1,110,235,0,1,153,0,0,2,0,2]]

classifier = KNeighborsClassifier()
classifier.fit(X,Y)
classifier_predict = classifier.predict(test_data)

print("prediction")
print(classifier_predict)
```

SampleLM.py

```
import pandas as pd
from sklearn.linear_model import LinearRegression
```

```
balance_data=pd.read_csv('C:/Users/Admin/PycharmProjects/hrflm/dataset.csv',sep=',',
header=None)

X= balance_data.values[1:, 0:13].reshape(-1, 1)
Y = balance_data.values[1:, 13].reshape(-1, 1)

test_data=[[62,0,0,140,268,0,0,160,0,3.6,0,2,2]]

linear_regressor = LinearRegression()
linear_regressor.fit(X,Y)
Y_pred = linear_regressor.predict(test_data)

print("prediction")
print(Y_pred)
```

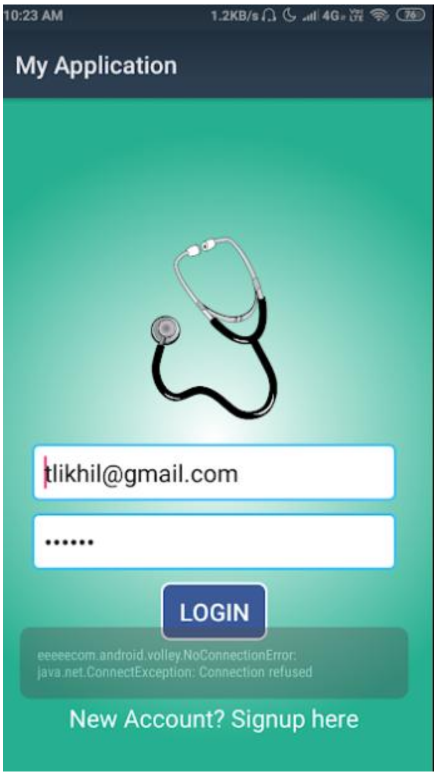
SampleNB.py

```
import pandas as pd
from sklearn.naive_bayes import GaussianNB
balance_data=pd.read_csv('C:/Users/Admin/PycharmProjects/hrflm/dataset.csv',sep=',',
header=None)

X= balance_data.values[1:, 0:13]
Y = balance_data.values[1:, 13]

test_data=[[62,0,0,140,268,0,0,160,0,3.6,0,2,2]]
model = GaussianNB()
model.fit(X,Y)
model_predict = model.predict(test_data)
print("prediction")
print(model_predict)
```

APPENDIX B

SCREENSHOTS

Login Page

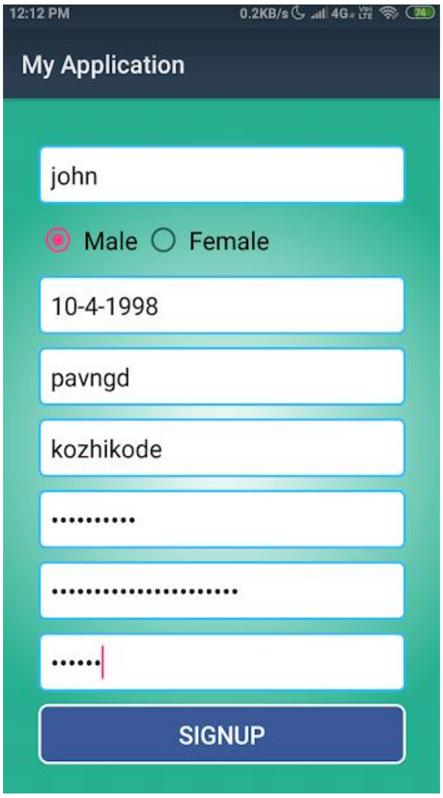


tlikhil@gmail.com

••••••

eeeeecom.android.volley.NoConnectionError: java.net.ConnectException: Connection refused

Signup Page



john

● Male ○ Female

10-4-1998

pavngd

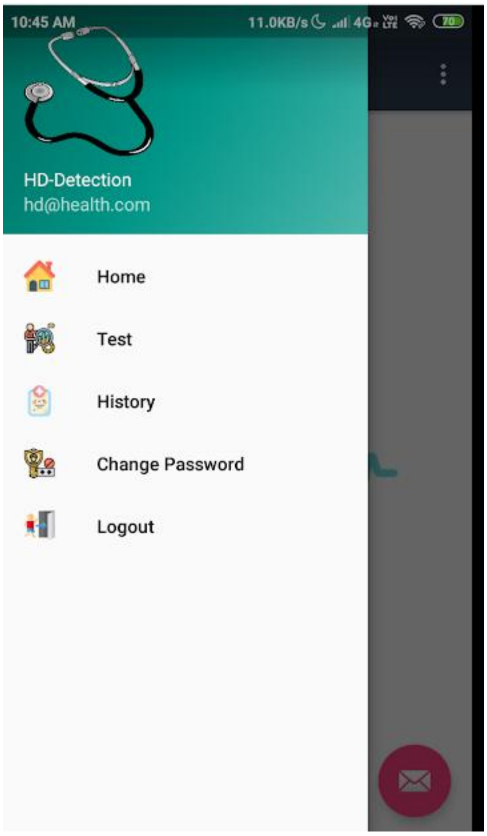
kozhikode

••••••••••

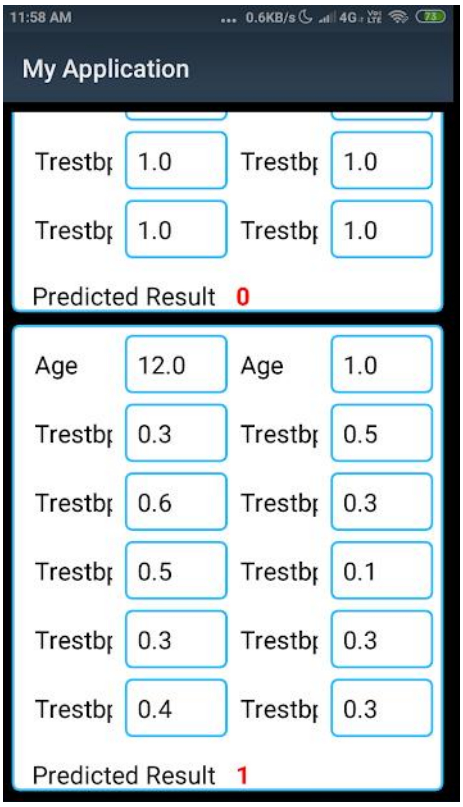
•••••••••••••••••••••

••••••

Home Page



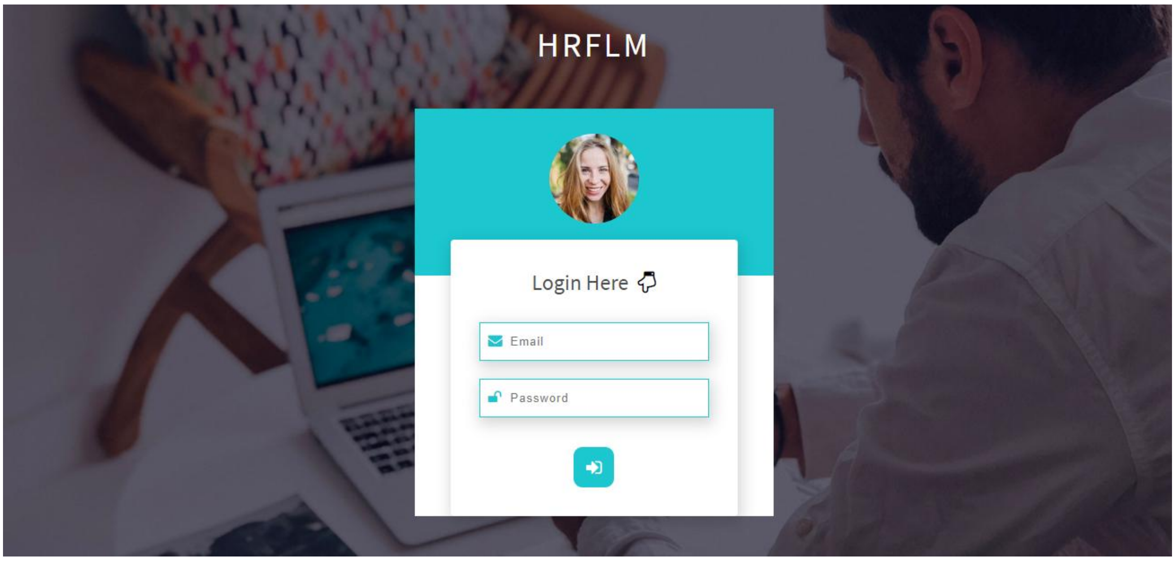
Prediction



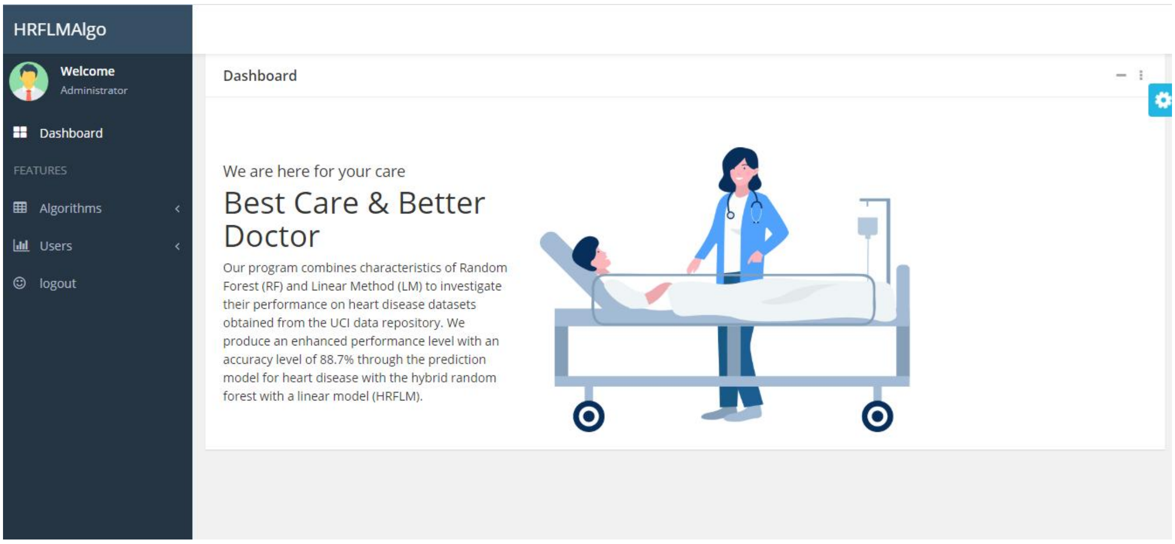
Trestbp 1.0 Trestbp 1.0
Trestbp 1.0 Trestbp 1.0
Predicted Result 0

Age 12.0 Age 1.0
Trestbp 0.3 Trestbp 0.5
Trestbp 0.6 Trestbp 0.3
Trestbp 0.5 Trestbp 0.1
Trestbp 0.3 Trestbp 0.3
Trestbp 0.4 Trestbp 0.3
Predicted Result 1

Login Page



Dashboard

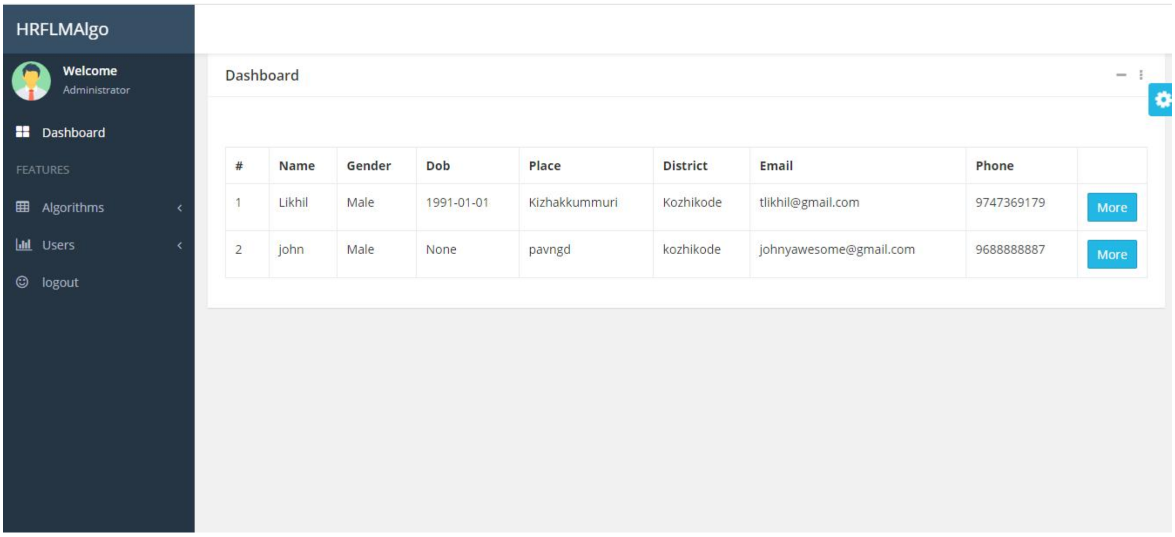


We are here for your care

Best Care & Better Doctor

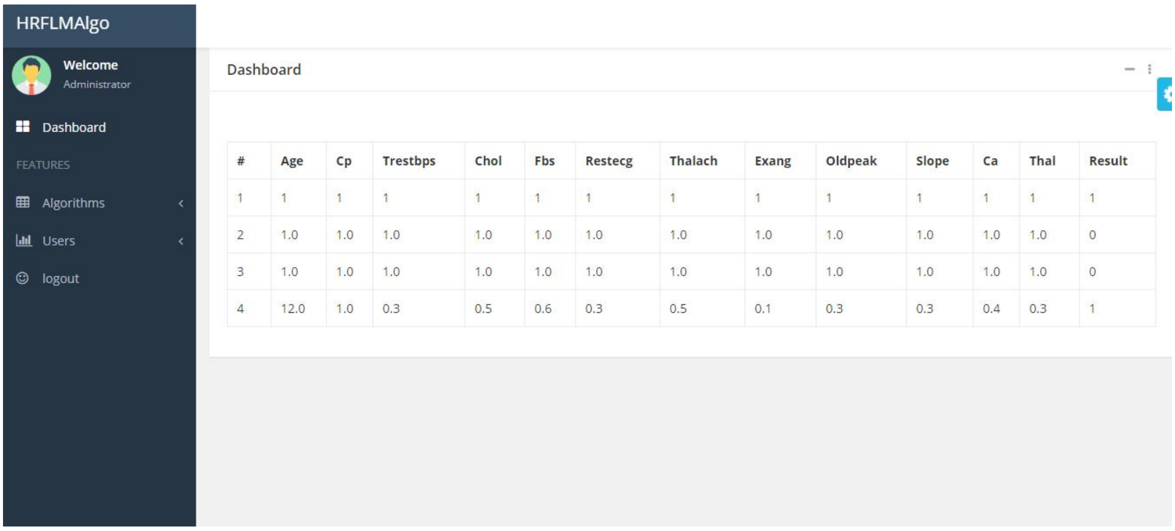
Our program combines characteristics of Random Forest (RF) and Linear Method (LM) to investigate their performance on heart disease datasets obtained from the UCI data repository. We produce an enhanced performance level with an accuracy level of 88.7% through the prediction model for heart disease with the hybrid random forest with a linear model (HRFLM).

Patient list



#	Name	Gender	Dob	Place	District	Email	Phone	
1	Likhil	Male	1991-01-01	Kizhakkummuri	Kozhikode	tlikhil@gmail.com	9747369179	More
2	john	Male	None	pavngd	kozhikode	johnyawesome@gmail.com	9688888887	More

Predictions



#	Age	Cp	Trestbps	Chol	Fbs	Restecg	Thalach	Exang	Oldpeak	Slope	Ca	Thal	Result
1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0
3	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0
4	12.0	1.0	0.3	0.5	0.6	0.3	0.5	0.1	0.3	0.3	0.4	0.3	1